ACL 2007

Proceedings of the Workshop on Balto-Slavonic Natural Language Processing 2007 Special Theme: Information Extraction and Enabling Technologies

June 29, 2007
Prague, Czech Republic

The Association for Computational Linguistics

Production and Manufacturing by
Omnipress
2600 Anderson Street
Madison, WI 53704
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Preface

There are over 400 million speakers of **Balto-Slavonic languages** world-wide (synonymously used: Balto-Slavic). As of 2007, almost a third of the 23 official European Union languages are Balto-Slavonic, i.e. Bulgarian, Czech, Latvian, Lithuanian, Polish, Slovak and Slovene. The two most recent rounds of the EU Enlargement fundamentally raised the interest in these languages: translators and interpreters for new language pairs need to be found, the interest in Machine (Aided) Translation systems has risen and tools that help language specialists and information-seeking individuals are now highly sought after.

For some of the countries speaking Balto-Slavonic languages, there is a **rich linguistic heritage**, and computational linguistics research and development is rather advanced. For others, however, there has been little development. This is due to the often small number of speakers of that language (Latvian, for instance, is spoken natively by around 1.5 Million people) combined with a lack of access to basic resources needed for Natural Language Processing such as machine-readable corpora and dictionaries, morphological analysers, part-of-speech taggers, parsers, etc. This leads to a linguistic **brain drain** as some of the best computational linguistics students go abroad or – even when staying in their mother country – work on developing new systems for English, French or German because resources for these languages are readily available.

Even when linguistic resources and tools are available to the scientific community, methods that have been successfully applied to Germanic and Romance languages cannot simply be ported to languages from the Balto-Slavonic group. The most well-known **linguistic phenomena** making Balto-Slavonic text analysis harder are the highly inflectional character and the related phenomenon of free word order in these languages. The invited speaker at the workshop, Adam Przepiórkowski from the Polish Academy of Sciences, explains these and a number of further specific linguistic phenomena typical for this language group. Interestingly, he points out that the differences may not always make text analysis tasks harder, but that they can also make some tasks easier.

When proposing this workshop to the Association for Computational Linguistics, we knew that Language Technology for some of the languages is not very advanced and that, to date, not much work has been carried out in the area of Information Extraction. The **objective of this workshop** was thus to promote the work on Balto-Slavonic languages by helping scientists to describe and share their resources and to describe their efforts, hoping that the experiences of a few will be useful for many others. We did not expect to receive papers presenting highly novel approaches to Information Extraction, but rather an adaptation of known methods to new languages and solutions for specific challenges regarding the Balto-Slavonic group. We found, however, that the current work on applying known approaches to a different language type did produce some very interesting work.

Contents of these proceedings

We received in total 20 submissions, of which only eight were specifically about **Information Extraction**, i.e. named entity recognition or the extraction of definitions. One of these papers targeted a more high-level Information Extraction task: an application combining more than one entity to fill a scenario template. The other 12 papers fall under the category Enabling Technology, describing mostly morphological tools and resources, taggers, corpora, WordNet developments and topic segmentation of texts.

Each of the submissions was reviewed by at least three peers. Finally, we have accepted 9 papers for oral presentations (**acceptance rate of 45%**) and selected a further three for poster presentations. We want to use this opportunity to thank the Programme Committee for their thorough reviews and mostly extensive and useful comments, as well as for keeping to the deadlines.

We do not want to claim that the proceedings of this workshop exhaustively reflect existing work on Information Extraction for the targeted language group. The submissions we received nevertheless gave us an overview of the **current state-of-the-art** for the various languages. Most of the submitted papers covered work on the languages Polish, Czech and Bulgarian. Only individual papers concerned Lithuanian, Croatian, Serbian, Ukrainian and Russian. When selecting papers for acceptance, an important criterion for us was to cover various languages.

The **final workshop program** includes 6 papers specifically addressing Information Extraction while the other six discuss Enabling Technologies. All of the papers clearly show how Natural Language Processing differs for Balto-Slavonic languages, compared to Germanic, Romance or other languages.

Our appeal

During the organisation of this workshop, we saw that some very good text analysis work has been carried out for some of the Balto-Slavonic languages. However, we were also reminded that many good scientists cannot work on interesting and promising research subjects and high-level applications because they first need to create the necessary linguistic resources. We therefore want to appeal to all researchers in the community to make as many resources and tools available to their peers as they can. This will in the end benefit all the scientists and – in the long run – the country as a whole. In this spirit, the Joint Research Centre has compiled – and distributes for free – a paragraph-aligned and subject domain-classified parallel corpus in 22 languages (the *JRC-Acquis*). May this resource be useful for the communities working on less-widely spoken languages.

Ispra, Italy, May 2007

Jakub Piskorski
Hristo Tanev
Bruno Pouliquen
Ralf Steinberger

European Commission – Joint Research Centre

Organizers

Chairs:

Jakub Piskorski, Joint Research Centre, IPSC
Bruno Pouliquen, Joint Research Centre, IPSC
Ralf Steinberger, Joint Research Centre, IPSC
Hristo Tanev, Joint Research Centre, IPSC

Program Committee:

Kalina Bontcheva, University of Sheffield
Tomaž Erjavec, Jožef Stefan Institute
Vladislav Kuboň, Charles Univeristy Prague
Anna Kupść, Université Paris 3
Rūta Marcinkevičienė, Vytautas Magnus University, Kaunas
Agnieszka Mykowiecka, Polish Academy of Sciences
Jakub Piskorski, Joint Research Centre, IPSC
Bruno Pouliquen, Joint Research Centre, IPSC
Hristo Tanev, Joint Research Centre, IPSC
Marko Tadić, University of Zagreb
Agata Savary, University of Tours
Kiril Simov, Bulgarian Academy of Sciences
Wojciech Skut, Google Inc.
Ralf Steinberger, Joint Research Centre, IPSC
Duško Vitas, University of Beograd
Roman Yangarber, Univeristy of Helsinki

Program Committee Co-Chairs:

Jakub Piskorski, Joint Research Centre, IPSC
Hristo Tanev, Joint Research Centre, IPSC

Additional Reviewers:

Niraj Aswani
Jan Daciuk
Camelia Ignat
Mladen Kolar
Domen Marinčič
Diana Maynard

Invited Speaker:

Adam Przepiórkowski, Polish Academy of Sciences

Table of Contents

Conference Program

Friday, June 29, 2005

9:00–9:20	Opening Remarks
	Invited Talk:
9:20–10:20	*Slavic Information Extraction and Partial Parsing* Adam Przepiórkowski
	Session 1: Information Extraction
10:20–10:45	*Implementation of Croatian NERC System* Božo Bekavac and Marko Tadić
10:45–11:15	Morning Cofee Break
11:15–11:40	*A Language Independent Approach for Name Categorization and Discrimination* Zornitsa Kozareva, Sonia Vázquez and Andrés Montoyo
11:40–12:05	*Lemmatization of Polish Person Names* Jakub Piskorski, Marcin Sydow and Anna Kupść
12:05–12:30	*Automatic Processing of Diabetic Patients' Hospital Documentation* Małgorzata Marciniak and Agnieszka Mykowiecka
12:30–14:30	Lunch

Friday, June 29, 2005 (continued)

Session 2: Information Extraction and Enabling Technologies

14:30–14:55 *Towards the Automatic Extraction of Definitions in Slavic*
Adam Przepiórkowski, Łukasz Degórski, Miroslav Spousta, Kiril Simov, Petya Osenova, Lothar Lemnitzer, Vladislav Kuboň and Beata Wójtowicz

14:55–15:20 *Unsupervised Methods of Topical Text Segmentation for Polish*
Dominik Flejter, Karol Wieloch and Witold Abramowicz

15:20–15:45 *Multi-word Term Extraction for Bulgarian*
Svetla Koeva

15:45–16:15 Afternoon Break

Session 3: Enabling Technologies

16:15–16:40 *The Best of Two Worlds: Cooperation of Statistical and Rule-Based Taggers for Czech*
Drahomíra Spoustová, Jan Hajič, Jan Votrubec, Pavel Krbec and Pavel Květoň

16:40–17:05 *Derivational Relations in Czech WordNet*
Karel Pala and Dana Hlaváčková

Session 4: Poster Session

17:05–18:00 *Multilingual Word Sense Discrimination: A Comparative Cross-Linguistic Study*
Alla Rozovskaya and Richard Sproat

17:05–18:00 *Named Entity Recognition for Ukrainian: A Resource-Light Approach*
Sophia Katrenko and Pieter Adriaans

17:05–18:00 *Morphological Annotation of the Lithuanian Corpus*
Erika Rimkutė, Vidas Daudaravičius and Andrius Utka

Slavonic Information Extraction and Partial Parsing

Adam Przepiórkowski
Insitute of Computer Science
Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland
adamp@ipipan.waw.pl

Abstract

Information Extraction (IE) often involves some amount of partial syntactic processing. This is clear in cases of interesting high-level IE tasks, such as finding information about who did what to whom (when, where, how and why), but it is also true in case of simpler IE tasks, such as finding company names in texts. The aim of this paper is to give an overview of Slavonic phenomena which pose particular problems for IE and partial parsing, and some phenomena which seem easier to treat in Slavonic than in Germanic or Romance; I also mention various tools which have been used for the partial processing of Slavonic.

1 Introduction

The aim of this paper is to give a general but rather biased overview of the problems of Information Extraction (IE) in Slavonic. In particular, I discuss linguistic phenomena which make IE in Slavonic harder than in Germanic or Romance, §2, but also those which seem to make it easier, §3. We will also look at various general tools which have been used in IE tasks in the context of Slavonic languages, especially at tools for partial (or shallow) parsing, §4.

I deal mainly with Polish, as a good representative of the Slavonic family: although Polish is a relatively large language, with about 44 million native speakers world-wide (over 38 million in Poland), the availability of linguistic resources and tools for this language does not reflect this fact: it compares unfavourably with Czech, and probably favourably with, say, Ukrainian.

2 Slavonic is Hard

There are various characteristics of Slavonic languages[1] that make them more difficult for automatic processing, whether shallow or deep, than Germanic and Romance languages.[2] The two of them which are most conspicuous, and identified as most problematic, e.g., in (Collins et al., 1999), are rich nominal inflection (§2.1) and free word order (§2.6). Others, causing problems to varying extents, include: idiosyncratic inflection of Slavonic proper names (§2.2); unstable inflection of some foreign names (§2.3); high degree of trans- and, especially, intra-paradigmatic syncretisms (§2.4); and, on the more syntactic level, the infamous quirkiness of numeral phrases (NumPs; §2.5).

2.1 Rich Nominal Inflection

The rich nominal inflection of Slavonic makes already the most basic IE task, namely Named Entity Recognition (NER), more difficult than in Germanic or Romance. Slavonic nouns, apart from in-

[1] Many of the typological features discussed below distinguish between, on the one hand, East Slavonic (Russian, Ukrainian, Belorussian, Rusyn), West Slavonic (Czech, Slovak, Upper and Lower Sorbian, Polish, Kashubian) and the Western subgroup of South Slavonic (Croatian, Bosnian, Serbian, Slovenian), and, on the other hand, the Eastern subgroup of South Slavonic (Bulgarian and Macedonian). In this and the next section I concentrate on the former group of Slavonic languages.

[2] By shallow or, equivalently, partial processing, I mean the task of finding *some* syntactic structure *without* using lexical resources such as valence dictionaries; by contrast, deep processing involves finding the *complete* sentence structures *with* the use of such lexical resources.

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 1–10,
Prague, June 2007. ©2007 Association for Computational Linguistics

flecting for number (singular and plural; in Slovenian and Sorbian also dual), famously inflect for about 6 (e.g., Russian, Slovenian) or 7 (e.g., Czech, Croatian, Polish, Ukrainian) cases: the exact number of cases cited in the literature for any particular language often depends on the granularity of description, so Belorussian and Slovak have either 6 or 7 cases, depending on the inclusion in the description the rare vocative forms, among the 7 Serbian cases, dative and locative are sometimes conflated because they "only" differ in accent, the Polish case system may be extended to 8 cases by postulating the distributive case (Gruszczyński, 1989, p. 89), while the number of Russian cases may also be reasonably increased to 8 by adding a second genitive and a second locative case (Jakobson, 1958).

While for many European languages a dictionary of lemmata of proper names is sufficient for the task of NER, (Steinberger and Pouliquen, 2007, §3.3) note that "a minimum of morphological treatment" is required for languages with rich nominal inflection, such as Balto-Slavonic or Finno-Ugric languages. Unfortunately, for the majority of Slavonic languages, there are no (freely) publicly available resources that could provide such "minimum morphological treatment" of proper names. For example, the only large free (but not open source) morphological analyser for Polish, Morfeusz (Woliński, 2006), contains very few proper names.[3] Moreover, the NE content of commercial analysers is often rather low, so that simple resource-light heuristics sometimes give better results (Urbańska and Mykowiecka, 2005, p. 214). Such heuristics usually involve the creation of inflected forms by adding typical suffixes (Popov et al., 2004; Urbańska and Mykowiecka, 2005; Steinberger and Pouliquen, 2007), where the suffix addition/substitution rules are either hand-generated (Urbańska and Mykowiecka, 2005) or automatically acquired (Steinberger and Pouliquen, 2007).

2.2 Different Inflection of Homonymous Common and Proper Nouns

As mentioned in (Piskorski, 2005) and discussed at length in (Piskorski et al., 2007b), many Polish surnames have the same base forms as common names, for example, GRZYB (lit. 'mushroom'), GOŁĄB (lit. 'pigeon') or KOWALSKI (lit. an adjective from 'smith'). This is a problem in itself in recognising proper names, but it is further exacerbated by the fact that such proper nouns may have different gender values, and different inflectional paradigms, than the corresponding common nouns. For example, while the common nouns GRZYB and GOŁĄB are, respectively, inanimate masculine and animate masculine (cf. fn. 5), the corresponding surnames are virile or feminine, depending on the denotation; in case of singular feminine names, they would not overtly inflect at all, while in case of singular masculine or plural uses, the forms are often different than corresponding common forms, e.g., the accusative singular and plural forms of GOŁĄB would be *gołębia* and *gołębie*, when used as a common noun, and *Gołąba* and *Gołąbów*, when used as a surname, etc. Obviously, once properly described, such inflectional differences may actually help in NER.

2.3 Difficult Inflection of Foreign Names

A problem relatively minor in comparison to other problems discussed here is the inflection of foreign names: although it is governed by strict prescriptive rules, native speakers are often unaware of them and different variants of the same form may be encountered in text; for example, while in Polish the correct spelling of the singular instrumental form of LINUX is *Linuksem*, the variant *Linuxem* is at least as common, and the starkly wrong *Linux'em* and *Linux-em* are also quite frequent. Similarly, probably few Poles realise that the correct locative forms of BRANDT and PEIRCE are *Brandcie* and *Peirsie*, and not, say, *Brandtcie* and *Peirce'ie*, and that although the locative of REMARQUE is *Remarque'u*, the instrumental is *Remarkiem*.[4] A comprehensive NER should be able to deal with various incorrect forms of foreign NE occurring in Slavonic texts.

On the other hand, the inflection of proper names

[3] A new version of Morfeusz, containing a large dictionary of proper names, is being prepared, but it is currently not clear if it is going to be freely available for non-commercial research purposes (M. Woliński, p.c.).

[4] See `http://so.pwn.pl/`.

depends on their pronunciation, i.e., on their origin. For example, the genitive of CHARLES is either *Charlesa* or *Charles'a*, depending on whether it is an English name or a French name. Another example, from (Piskorski et al., 2007b), is WILDE, whose genitive form is either *Wilde'a* (English) or *Wildego* (German). This feature, when properly encoded, may actually help distinguish between entities in NER.

2.4 Tagset Size and Syncretisms

A rich inflection system also implies that the size of the tagset is very large. For example, given that a Polish nominal form may have one of 2 numbers, one of 7 cases and one of 5 genders,[5] there are 70 possible nominal tags, not counting gerundial and pronominal forms. In fact, there are 4179 possible tags in the IPI PAN Tagset of Polish (Przepiórkowski and Woliński, 2003a; Przepiórkowski and Woliński, 2003b), of which around 1150 occur in nature (Przepiórkowski, 2006b). Similarly, sizes of Czech tagsets range from 1171 (Hajič and Hladká, 1997), through 1631 (Pala et al., 1998), to theoretically 4257, but "only" about 1100 actually used (Mirovský et al., 2002). Such detailed tagsets make it difficult to reach high accuracy, which — on the assumption that syntactic parsing is preceded by full morphosyntactic disambiguation — has negative influence on syntactic processing.

Another problem connected to the rich inflection system of Slavonic languages is the large number of syncretisms. For example, a typical Polish adjective may have 11 textually different forms (e.g., for BIAŁY 'white': *biali*, *biała*, *białą*, *białe*, *białego*, *białej*, *białemu*, *biały*, *białych*, *białym*, *białymi*), but as many as 70 different tags (2 numbers × 7 cases × 5 genders). There are also various systematic nominal syncretisms which to some extent annul the advantages that rich case system presents for the identification of grammatical roles. For example, in plural, Polish non-virile (non-human-masculine) nouns have the same form in the nominative and in the accusative, while in the singular, inanimate masculine and neuter forms do. Similarly, virile and animate masculine nouns have the same singular accusative and singular genitive forms. So, for example, in the rather artificial sentence *Samochody dwie minuty wyprzedzają autobusy* '(The) cars (for) two minutes are overtaking (the) buses', each of *samochody*, *dwie minuty* and *autobusy* may be interpreted as either nominative or accusative, i.e., as the subject (nominative), the object (accusative) or a temporal adjunct (accusative).

2.5 Numeral Phrases

An area of Slavonic syntax very well-known in theoretical linguistics is the syntactic behaviour of NumPs (Corbett, 1978; Franks, 1995); numerals also turn out to be awkward for automatic processing in various ways.[6]

First, the case of the noun (phrase) within an NumP depends on the numeral[7] and on the position of the whole NumP in the sentence. For example, for NumPs in the subject position, the noun is in the nominative case, roughly, if the numeral is or ends in 2, 3 or 4 (with the exception of 12, 13 and 14), and it is genitive otherwise.[8] This means that the shallow processor should recognise as a possible currency quantity the sequence *152 dolary* and *155 dolarów*, but not **152 dolarów* or **155 dolary*.[9]

Second, in case of "typical" numerals (not ending in 2, 3 or 4), the Polish NumP in subject position does *not* agree with the verb; instead, the verb occurs in the default 3rd person singular neuter form,[10]

[5] Traditionally, 3 genders were assumed for Polish, as for many other European languages, but (Mańczak, 1956) conclusively shows that at least 5 gender values must be adopted in Polish: *virile* (called also *m1*, *personal masculine* and *human masculine*), *animate masculine* (*m2*), *inanimate masculine* (*m3*), *neuter* and *feminine*. Although this repertoire of genders was only recently adopted in general dictionaries (Bańko, 2000), it is still rather conservative; e.g., (Saloni, 1976) proposes 9 genders.

[6] One of the largest formal grammars of Polish, (Świdziński, 1992), implemented as a wide coverage deep parser in (Woliński, 2004), does not deal with NumPs at all. Later modifications of the parser in (Ogrodniczuk, 2006) include some limited treatment of numerals.

[7] This property turned out to be problematic for adapting the GF Parallel Resource Grammar to Russian (Khegai, 2006).

[8] Another exception is JEDEN '1', which is actually an adjective, rather than a numeral (Przepiórkowski, 2006a). Also, the description above holds for non-virile genders, but is even more complicated for virile.

[9] The latter may occur in contexts like: ... *według paragrafu 155 dolary nie są środkiem płatniczym w Polsce* '... according to paragraph 155 dollars are not a valid currency in Poland'.

[10] I argue elsewhere (Przepiórkowski, 1996; Przepiórkowski, 2004b) that such NumPs in subject position actually bear the accusative case; hence, the lack of agreement.

which may make discovering the subject-verb relation more difficult.

Finally, and rather marginally, "typical" NumPs in copular constructions trigger very atypical agreement with the predicative adjective, e.g.: *40 głosów było nieważnych/nieważne* '40 votes be-3RD.SG.NEUT invalid-PL.GEN/ACC'. It is easy to overlook such constructions when developing a shallow grammar, and — since they are rare — it is difficult to learn them automatically from corpora.

2.6 Free Word Order

Last but certainly not least, the relatively free word order[11] makes the discovering of who did what to whom (when, where, how and why) much more difficult than finding the relative order of NPs and PPs in the sentence. It may seem that the rich case system may help here, as — with active forms of verbs — subjects are usually nominative and objects are often accusative, but matters are much more complicated because of the widespread syncretisms mentioned in §2.4, esp. the systematic nominative-accusative and accusative-genitive syncretisms, and because both complements and adjuncts may be expressed by the same cases (e.g., accusative temporal adjuncts may look like objects of transitive verbs).

While the relatively free word order is seriously felt in deep parsers and leads to the multiplication of analyses, to the best of our knowledge most IE work in Slavonic to date has concentrated on lower-level tasks such as NER and, hence, has not yet tried to systematically deal with this problem.

3 Slavonic is Easy

On a more positive note, the rich Slavonic inflectional system may help at the higher levels of processing. There are various linguistic phenomena where overt case, gender and number agreement allows to differentiate between interpretations and, hence, to extract the information about who did what to whom. To give two trivial constructed examples: the English sentence *I saw him drunk* is ambiguous in ways that are necessarily disambiguated by the two Polish translations of that sentence: *Widziałem go pijany* '(I) saw him drunk-NOM' and *Widziałem go pijanego* '(I) saw him drunk-ACC'. Perhaps more interestingly, the lexical aspect of Slavonic verbs may make conspicuous the meanings which are only implicit in other languages, as in the Polish *Skoczył na stół* '(He) jumped-PERF on (the) table-ACC' versus *Skakał na stole* '(He) jumped-IMPERF on (the) table-LOC', both translated into the English *He jumped on the table*.

One phenomenon important for high level IE where the rich inflectional system plays a positive role, however, is coordination.

Coordination is infamous both in theoretical linguistics and in Natural Language Processing (NLP); in fact, while recent years witnessed an increase of theoretical linguistic works on various aspects of coordination, it seems that NLP lags behind in addressing this phenomenon head on. One of the exceptions is (Dale and Mazur, 2007), which deals with the problem of identifying the number of Named Entities (NEs) in expressions of the form "X and Y", where X and Y are sequences of capitalised words, e.g.: "Victorian Casino and Gaming Authority" (single entity) or "American Express and Visa International" (two entities). (Dale and Mazur, 2007) note that the problem is statistically non-negligible, as around 5.7% of sequences of capitalised words with an optional conjunction (i.e., candidates for NEs) actually contain a conjunction. Similarly, (Rus et al., 2007, p. 229) discuss the bracketing problem in phrases such as "[soccer and tennis] player" and "navy and [marine corps]", noting that "[p]arsing base Noun Phrases ... is not handled by current state-of-the-art syntactic parsers". Another kind of coordination ambiguity is considered in (Steiner, 2006), namely, the "NP and NP" sequence as either an NP-coordination, or a part of sentential coordination (where the first NP is an object of the preceding verb and the second NP is the subject of the following verb).

Slavonic rich inflection makes the processing of such potentially coordinate structures easier. For example, case disagreement between two apparently coordinated NPs is a strong clue that they in fact belong to separate coordinated clauses, while agreement is a (perhaps weaker) clue that they form an ac-

[11] Of course, the term *free word order* as applied to Slavonic means that the word order is conditioned largely by information structure (i.e., not really free); modelling the constraints of information structure on word order is particularly important in text generation (Kruijff-Korbayová and Kruijff, 1999).

tually coordinated NP.[12] Similarly, (dis)agreement in case, number and gender may help decide whether two apparently coordinated adjectival forms actually form a coordinate structure.

4 Slavonic is Processable

After discussing ways in which Slavonic languages seem to be hard or easy for Information Extraction, let us look at practical attempts at Slavonic IE, especially those involving partial parsing.

It seems that there have been relatively few attempts at applying shallow (or partial; cf. fn. 2) grammars to particular practical tasks. In some of these attempts no particular dedicated language processing system was used to implement shallow grammars: apparently they were coded directly in the host programming language.

One example is (Sharoff, 2004), where shallow parsing is used for the identification of prepositional Multi Word Expressions in Russian, with the following explanation of reasons for performing some language-dependent processing: "Given that the word order in Russian (and other Slavonic languages) is relatively free and a typical word (i.e. lemma) has many forms (typically from 9 for nouns to 50 for verbs), the sequences of exact N-grams are much less frequent than in English, thus rendering purely statistical approaches useless."

For Polish, simple shallow grammars were implemented for the tasks of question answering (Piechociński and Mykowiecka, 2005) and automatic valence acquisition (Fast and Przepiórkowski, 2005; Przepiórkowski and Fast, 2005); in the latter case a grammar was implemented as a cascade of Perl regular expressions. Similarly, (Zeman, 2001) describes a Perl regular expression implementation of a shallow preprocessor for a deep statistical parser. Much earlier, (Nenadić and Vitas, 1998; Nenadić, 2000) developed shallow grammars of Serbo-Croatian for the recognition of noun phrases (NPs) and certain kinds of coordinate structures. See also (Bekavac and Tadić, 2007) on the recognition of Croatian NEs with regular grammars.

Moreover, for Bulgarian a more general integrated system was developed, called LINGUA (Tanev and Mitkov, 2002), which — apart from modules for tokenisation, morphosyntactic analysis and disambiguation, and anaphora resolution — includes an NP extractor and a bottom-up grammar of Bulgarian. This system, together with a set of shallow patterns for identifying definition patterns, has been employed in a Question Answering prototype system (Tanev, 2004). Bulgarian pattern-matching grammars are also employed in (Koeva, 2007).

Apart from these language-specific implementations, there exist tools and toolboxes which facilitate various IE tasks, including shallow parsing. Probably the best known such a general system is GATE (Cunningham et al., 1995; Cunningham et al., 2002), which contains some NE resources for Bulgarian and Russian (Humphreys et al., 2002; Popov et al., 2004) and allows to write shallow (regular) grammars in the JAPE subsystem (Cunningham et al., 2000).

A system similar in scope is SProUT (Becker et al., 2002), whose shallow parsing language allows to write regular grammars over HPSG-style (Pollard and Sag, 1994) typed feature structures and which includes the operation of unification. Preliminary work on adapting SProUT to the processing of Baltic and Slavonic languages is presented in (Drożdżyński et al., 2003), with much subsequent work devoted to the processing of Polish, especially, in the area of Information Extraction from medical texts (Piskorski et al., 2004; Piskorski, 2004a; Piskorski, 2004b; Marciniak et al., 2005; Mykowiecka et al., 2005a; Mykowiecka et al., 2005b; Marciniak and Mykowiecka, 2007).

Although GATE and SProUT may be adapted to the processing of XML documents, they are perhaps not the most natural choice for the further processing of morphosyntactically annotated documents in, for example, the XCES (XML Corpus Encoding Standard; (Ide et al., 2000)) format, as assumed, e.g., in the IPI PAN Corpus of Polish (Przepiórkowski, 2004a), in the Slovak National Corpus (Garabík and Gianitsová-Ološtiaková, 2005), or in the LT4eL project (`http://www.lt4el.eu/`). Specialised XML-aware tools exist for such tasks.

One of the earliest collections of XML processing tools is the LT XML library (Brew et al., 2000), whose second edition, LT-XML2 is currently under preparation. One of the tools in that new edition, `lxtransduce` (Tobin, 2005), is an efficient pro-

[12] Again, this test may fail due to case syncretisms; cf. §2.4.

gram to add mark-up to XML files via regular grammars over XML elements; this tool is currently used for implementing definition-extraction grammars for Bulgarian, Czech and Polish (Przepiórkowski et al., 2007).

A system well-known in Slavonic NLP is CLaRK (Simov et al., 2001; Simov et al., 2002); it implements various XML mechanism and proposes a language for developing shallow grammars over XML documents; such grammars have been implemented for Bulgarian, as reported in (Simov et al., 2004; Simov and Osenova, 2004).

Finally, a new system, SPADE (*Shallow Parsing and Disambiguation Engine*), abbreviated to "♠" (Unicode character 0x2660), has recently been developed at the Institute of Computer Science, Polish Academy of Sciences (Przepiórkowski, 2007b; Buczyński, 2007). This tool, unlike many other shallow parsing tools,[13] accepts a possibly morphosyntactically ambiguous (XCES-encoded) input and performs simultaneous morphosyntactic disambiguation and shallow parsing. For example, the rule below, called `P + co/kto`, will match a possible preposition followed by a possible form of one of the pronouns CO 'what' or KTO 'who',[14] it will try to unify the selected case of the preposition with the case of the pronoun and, if that succeeds, it will mark any non-unified interpretations as rejected and it will mark the two words as a prepositional group with the preposition (cf. `1` below) as its syntactic head and the pronoun (cf. `2`) as its semantic head.[15] Moreover, any non-prepositional interpretations of the first segment of the match and any non-nominal interpretations of the second segment will be marked as incorrect. The language for specification of segments is based on the query syntax of the Poliqarp corpus search engine (Przepiórkowski et al., 2004; Janus and Przepiórkowski, 2007), in turn based on CQP (Christ, 1994).

```
RULE P + co/kto

Match: [pos~"prep"][base~"co|kto"]
Cond:  unify(case,1,2)
Synt:  group(PrepNG,1,2)
Morph: leave(pos~~"prep",1)
Morph: leave(pos~~"subst",2)
```

SPADE is currently employed for the shallow processing of the IPI PAN Corpus of Polish.

5 Conclusion

While the relatively free word order of Slavonic languages makes the processing of Slavonic unambiguously harder, I claim that the effects of the rich nominal inflection are mixed: rich inflection dramatically increases the complexity of low-level IE tasks such as NER, but it is beneficial for high-level IE tasks which involve filling scenario templates, as it facilitates identifying grammatical roles, parsing coordination, etc. Moreover, as becomes clear on the basis of the overview of practical work on Slavonic IE in the last decade, recent years have witnessed substantially increased interest and activity in the area. I am convinced that the Balto-Slavonic Natural Language Processing workshop at ACL 2007 will further catalyse the development of this field.

Acknowledgments

I am grateful to Jakub Piskorski, Bruno Pouliquen, Ralf Steinberger and Hristo Tanev for their generous invitation to the BSNLP workshop, and to the European Commission's Joint Research Centre in Ispra, Italy, for financial support. This paper was written during my stay at the University of Regensburg, Germany, within the DAAD-MNiSW cooperation programme *Automatic methods of extracting linguistic knowledge from corpora* (2006–2007). Many thanks to Qba Piskorski and Roland Meyer for comments on the (as it turned out, pre-) final versions of the paper.

References

Mirosław Bańko, editor. 2000. *Inny słownik języka polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Markus Becker, Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer, and Feiyu Xu. 2002. SProUT — shallow processing with typed feature structures and unification. In *Proceedings of the International Conference on NLP (ICON 2002)*, Mumbai, India.

[13] But the shallow grammars for Serbo-Croatian described in (Nenadić and Vitas, 1998; Nenadić, 2000) were developed with similar goals in mind.

[14] Left and right context of a match may be specified; here they are empty.

[15] A rationale for distinguishing these two kinds of heads is given in (Przepiórkowski, 2007a).

Božo Bekavac and Marko Tadić. 2007. Implementation of Croatian NERC system. In Piskorski et al. (Piskorski et al., 2007a).

Leonard Bolc, Zbigniew Michalewicz, and Toyoaki Nishida, editors. 2005. *Intelligent Media Technology for Communicative Intelligence, Second International Workshop, IMTCI 2004, Warsaw, Poland, September 13-14, 2004, Revised Selected Papers*, volume 3490 of *Lecture Notes in Computer Science*. Springer-Verlag.

Chris Brew, David McKelvie, Richard Tobin, Henry Thompson, and Andrei Mikheev, 2000. *The XML Library LT XML version 1.2: User documentation and reference guide*. Language Technology Group, University of Edinburgh. `http://www.ltg.ed.ac.uk/software/xml/xmldoc/xmldoc.html`.

Aleksander Buczyński. 2007. An implementation of combined partial parser and morphosyntactic disambiguator. In *Proceedings of ACL 2007 Student Research Workshop*.

Oli Christ. 1994. A modular and flexible architecture for an integrated corpus query system. In *COMPLEX'94*, Budapest.

Michael Collins, Jan Hajič, Lance Ramshaw, and Christoph Tillmann. 1999. A statistical parser for Czech. In *Proceedings of ACL 1999*, pages 505–518, University of Maryland.

Greville G. Corbett. 1978. Numerous squishes and squishy numerals in Slavonic. In Bernard Comrie, editor, *Classification of Grammatical Categories*, pages 43–73. Linguistic Research, Inc., Edmonton.

Hamish Cunningham, Robert Gaizauskas, and Yorick Wilks. 1995. A general architecture for text engineering (GATE) — a new approach to language engineering R&D. Technical report, Department of Computer Science, University of Sheffield. `http://xxx.lanl.gov/abs/cs.CL/9601009`.

Hamish Cunningham, Diana Maynard, and Valentin Tablan. 2000. JAPE: a Java Annotation Patterns Engine (second edition). Technical Report CS–00–10, Department of Computer Science, University of Sheffield.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Robert Dale and Paweł Mazur. 2007. Handling conjunctions in named entities. In Gelbukh (Gelbukh, 2007), pages 131–142.

Witold Drożdżyński, Petr Homola, Jakub Piskorski, and Vytautas Zinkevičius. 2003. Adapting SProUT to processing Baltic and Slavonic languages. In Hamish Cunningham, E. Paskaleva, Kalina Bontcheva, and G. Angelova, editors, *Information Extraction for Slavonic and Other Central and Eastern European Languages*, pages 18–25, Borovets, Bulgaria.

ELRA. 2004. *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, Lisbon.

Jakub Fast and Adam Przepiórkowski. 2005. Automatic extraction of Polish verb subcategorization: An evaluation of common statistics. In Vetulani (Vetulani, 2005), pages 191–195.

Steven Franks. 1995. *Parameters of Slavic Morphosyntax*. Oxford University Press, New York.

Radovan Garabík, editor. 2005. *Computer Treatment of Slavic and East European Languages: Proceedings of the Third International Seminar, Bratislava, Slovakia, 10–12 November 2005*, Bratislava. VEDA: Vydavatel'stvo Slovenskej akadéme vied.

Radovan Garabík and Lucia Gianitsová-Ološtiaková. 2005. Manual morphological annotation of Slovak translation of Orwell's novel 1984 — methods and findings. In Garabík (Garabík, 2005), pages 59–66.

Alexander Gelbukh, editor. 2007. *Computational Linguistics and Intelligent Text Processing (CICLing 2007)*, Lecture Notes in Computer Science, Berlin. Springer-Verlag.

Włodzimierz Gruszczyński. 1989. *Fleksja Rzeczowników Pospolitych we Współczesnej Polszczyźnie Pisanej (na materiale „Słownika języka polskiego" PAN pod redakcją W. Doroszewskiego)*, volume 122 of *Prace Językoznawcze*. Ossolineum, Wrocław.

Jan Hajič and Barbara Hladká. 1997. Probabilistic and rule-based tagger of an inflective language - a comparison. In *Proceedings of the ANLP'97*, pages 111–118, Washington, DC.

K. Humphreys, R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham, and Y. Wilks. 2002. Slavonic named entities in GATE. Research Memorandum CS-02-01, University of Sheffield.

Nancy Ide, Patrice Bonhomme, and Laurent Romary. 2000. XCES: An XML-based standard for linguistic corpora. In *Proceedings of the Linguistic Resources and Evaluation Conference*, pages 825–830, Athens, Greece.

Roman O. Jakobson. 1958. Morfologičeskie nabljudenija nad slavjanskim skloneniem. In *Selected Writings II*, pages 154–183. Mouton, The Hague.

Daniel Janus and Adam Przepiórkowski. 2007. Poliqarp: An open source corpus indexer and search engine with syntactic extensions. In *Proceedings of ACL 2007 Demo Session*.

Janna Khegai. 2006. GF parallel resource grammars and Russian. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 475–482, Sydney, Australia. Association for Computational Linguistics.

Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors. 2005. *Intelligent Information Processing and Web Mining*. Advances in Soft Computing. Springer-Verlag, Berlin.

Svetla Koeva. 2007. Multi-word term extraction for Bulgarian. In Piskorski et al. (Piskorski et al., 2007a).

Ivana Kruijff-Korbayová and Geert-Jan M. Kruijff. 1999. Handling word order in a multilingual system for generation of instructions. In Václav Matousek, Pavel Mautner, Jana Ocelíková, and Petr Sojka, editors, *Text, Speech and Dialogue - Second International Workshop, TSD'99, Plzen, Czech Republic, September 1999*, pages 83–88, Berlin. Springer-Verlag.

Witold Mańczak. 1956. Ile jest rodzajów w polskim? *Język Polski*, XXXVI(2):116–121.

Małgorzata Marciniak and Agnieszka Mykowiecka. 2007. Automatic processing of diabetic patients' hospital documentation. In Piskorski et al. (Piskorski et al., 2007a).

Małgorzata Marciniak, Agnieszka Mykowiecka, Anna Kupść, and Jakub Piskorski. 2005. Intelligent content extraction from Polish medical texts. In Bolc et al. (Bolc et al., 2005), pages 68–78.

Jiří Mirovský, Roman Ondruška, and Daniel Průša. 2002. Searching through Prague Dependency Treebank: Conception and architecture. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories (TLT2002)*, pages 114–122, Sozopol, Bulgaria.

Agnieszka Mykowiecka, Anna Kupść, and Małgorzata Marciniak. 2005a. Rule-based medical content extraction and classification. In Kłopotek et al. (Kłopotek et al., 2005), pages 237–246.

Agnieszka Mykowiecka, Małgorzata Marciniak, and Anna Kupść. 2005b. Making shallow look deeper: Anaphora and comparisons in medical information extraction. In Vetulani (Vetulani, 2005).

Goran Nenadić. 2000. Local grammars and parsing coordination of nouns in Serbo-Croatian. In *Proceedings of Text, Dialogue and Speech (TSD) 2000*, pages 57–62. Springer-Verlag.

Goran Nenadić and Duško Vitas. 1998. Using local grammars for agreement modeling in highly inflective languages. In *Proceedings of Text, Dialogue and Speech (TSD) 1998*, pages 91–96.

Maciej Ogrodniczuk. 2006. *Weryfikacja korpusu wypowiedników polskich (z wykorzystaniem gramatyki formalnej Świdzińskiego)*. Ph. D. dissertation, Warsaw University, Warsaw.

Karel Pala, Pavel Rychlý, and Pavel Smrž. 1998. Corpus annotation in inflectional languages: Czech. In A Min Tjoa and Roland R. Wagner, editors, *Ninth International Workshop on Database and Expert Systems Applications*, pages 149–153, Los Alamitos, California.

Dariusz Piechociński and Agnieszka Mykowiecka. 2005. Question answering in Polish using shallow parsing. In Garabík (Garabík, 2005), pages 167–173.

Jakub Piskorski. 2004a. Extraction of Polish named-entities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004* (ELR, 2004), pages 313–316.

Jakub Piskorski. 2004b. Rule-based named-entity recognition for Polish. In *Proceedings of the Workshop on Named-Entity Recognition for NLP Applications held in conjunction with the 1st International Joint Conference on NLP, March 2004*, Sanya, Hainan Island, China.

Jakub Piskorski. 2005. Named-entity recognition for Polish with SProUT. In Bolc et al. (Bolc et al., 2005).

Jakub Piskorski, Peter Homola, Małgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiórkowski, and Marcin Woliński. 2004. Information extraction for Polish using the SProUT platform. In Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent Information Processing and Web Mining*, Advances in Soft Computing, pages 227–236. Springer-Verlag, Berlin.

Jakub Piskorski, Bruno Pouliquen, Ralf Steinberger, and Hristo Tanev, editors. 2007a. *Proceedings of the BSNLP workshop at ACL 2007*, Prague.

Jakub Piskorski, Marcin Sydow, and Anna Kupść. 2007b. Lemmatization of Polish person names. In Piskorski et al. (Piskorski et al., 2007a).

Carl Pollard and Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago University Press / CSLI Publications, Chicago, IL.

Borislav Popov, Angel Kirilov, Diana Maynard, and Dimitar Manov. 2004. Creation of reusable components and language resources for Named Entity Recognition in Russian. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004* (ELR, 2004), pages 309–312.

Adam Przepiórkowski. 1996. Case assignment in Polish: Towards an HPSG analysis. In Claire Grover and Enric Vallduví, editors, *Studies in HPSG*, volume 12 of *Edinburgh Working Papers in Cognitive Science*, pages 191–228. Centre for Cognitive Science, University of Edinburgh.

Adam Przepiórkowski. 2004a. *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Adam Przepiórkowski. 2004b. O wartości przypadka podmiotów liczebnikowych. *Biuletyn Polskiego Towarzystwa Językoznawczego*, LX:133–143.

Adam Przepiórkowski. 2006a. O dystrybutywnym PO i liczebnikach jedynkowych. *Polonica*, XXVI–XXVII:171–178.

Adam Przepiórkowski. 2006b. The potential of the IPI PAN Corpus. *Poznań Studies in Contemporary Linguistics*, 41:31–48.

Adam Przepiórkowski. 2007a. On heads and coordination in valence acquisition. In Gelbukh (Gelbukh, 2007), pages 50–61.

Adam Przepiórkowski. 2007b. A preliminary formalism for simultaneous rule-based tagging and partial parsing. In Georg Rehm, Andreas Witt, and Lothar Lemnitzer, editors, *Data Structures for Linguistic Resources and Applications: Proceedings of the Biennial GLDV Conference 2007*, pages 81–90. Gunter Narr Verlag, Tübingen.

Adam Przepiórkowski and Jakub Fast. 2005. Baseline experiments in the extraction of Polish valence frames. In Kłopotek et al. (Kłopotek et al., 2005), pages 511–520.

Adam Przepiórkowski and Marcin Woliński. 2003a. A flexemic tagset for Polish. In *Proceedings of* Morphological Processing of Slavic Languages*, EACL 2003*, pages 33–40, Budapest.

Adam Przepiórkowski and Marcin Woliński. 2003b. The unbearable lightness of tagging: A case study in morphosyntactic tagging of Polish. In *Proceedings of the* 4th International Workshop on Linguistically Interpreted Corpora (LINC-03)*, EACL 2003*, pages 109–116.

Adam Przepiórkowski, Zygmunt Krynicki, Łukasz Dębowski, Marcin Woliński, Daniel Janus, and Piotr Bański. 2004. A search tool for corpora with positional tagsets and ambiguities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004* (ELR, 2004), pages 1235–1238.

Adam Przepiórkowski, Łukasz Degórski, Miroslav Spousta, Kiril Simov, Petya Osenova, Lothar Lemnitzer, Vladislav Kuboň, and Beata Wójtowicz. 2007. Towards the automatic extraction of definitions in Slavic. In Piskorski et al. (Piskorski et al., 2007a).

Vasile Rus, Sireesha Ravi, Mihai C. Lintean, and Philip M. McCarthy. 2007. Unsupervised method for parsing coordinated base noun phrases. In Gelbukh (Gelbukh, 2007), pages 229–240.

Zygmunt Saloni. 1976. Kategoria rodzaju we współczesnym języku polskim. In Roman Laskowski, editor, *Kategorie gramatyczne grup imiennych we współczesnym języku polskim*, volume 14 of *Prace Instytutu Języka Polskiego*, pages 43–78. Ossolineum, Wrocław.

Serge Sharoff. 2004. What is at stake: a case study of Russian expressions starting with a preposition. In Takaaki Tanaka, Aline Villavicencio, Francis Bond, and Anna Korhonen, editors, *Second ACL Workshop on Multiword Expressions: Integrating Processing*, pages 17–23, Barcelona, Spain. Association for Computational Linguistics.

Kiril Simov and Petya Osenova. 2004. A hybrid strategy for regular grammar parsing. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004* (ELR, 2004), pages 431–434.

Kiril Simov, Z. Peev, Milen Kouylekov, Alexander Simov, M. Dimitrov, and A. Kiryakov. 2001. CLaRK — an XML-based system for corpora development. In Paul Rayson, Andrew Wilson, Tony McEnery, Andrew Hardie, and Shereen Khoja, editors, *Proceedings of the Corpus Linguistics 2001 Conference*, pages 558–560, Lancaster.

Kiril Simov, Petya Osenova, Milena Slavcheva, Sia Kolkovska, Elisaveta Balabanova, Dimitar Doikoff, Krassimira Ivanova, Alexander Simov, and Milen Kouylekov. 2002. Building a linguistically interpreted corpus of Bulgarian. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002*, pages 1729–1736, Las Palmas, Canary Islands, Spain. ELRA.

Kiril Simov, Petya Osenova, Sia Kolkovska, Elisaveta Balabanova, and Dimitar Doikoff. 2004. A language resources infrastructure for Bulgarian. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004* (ELR, 2004), pages 1685–1688.

Ralf Steinberger and Bruno Pouliquen. 2007. Cross-lingual Named Entity Recognition. *Linguisticae Investigationes*. Special Issue on Named Entity Recognition and Categorisation, Satoshi Sekine and Elisabete Ranchhod (eds.), Forthcoming.

Ilona Steiner. 2006. Coordinate structures: On the relationship between parsing preferences and corpus frequencies. In *Pre-Proceedings of the International Conference on Linguistic Evidence: Empirical, Theoretical and Computational Perspectives, Tübingen, 2–4 February 2006*, pages 88–92, Tübingen. SFB 441 "Linguistic Data Structures", University of Tübingen, Germany.

Marek Świdziński. 1992. *Gramatyka formalna języka polskiego*, volume 349 of *Rozprawy Uniwersytetu Warszawskiego*. Wydawnictwa Uniwersytetu Warszawskiego, Warsaw.

Hristo Tanev. 2004. Socrates: A question answering prototype for Bulgarian. In *Recent Advances in Natural Language Processing III, Selected Papers from RANLP 2003*, pages 377–386. John Benjamins.

Hristo Tanev and Ruslan Mitkov. 2002. Shallow language processing architecture for Bulgarian. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002)*, Taipei.

Richard Tobin, 2005. *Lxtransduce, a replacement for fsgmatch*. University of Edinburgh. `http://www.cogsci.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html`.

Dominika Urbańska and Agnieszka Mykowiecka. 2005. Multi-words Named Entity Recognition in Polish texts. In Garabík (Garabík, 2005), pages 208–215.

Zygmunt Vetulani, editor. 2005. *Proceedings of the* 2nd Language & Technology Conference, Poznań, Poland.

Marcin Woliński. 2004. *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph. D. dissertation, Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Marcin Woliński. 2006. Morfeusz — a practical tool for the morphological analysis of Polish. In Mieczysław A. Kłopotek, Sławomir T. Wierzchoń, and Krzysztof Trojanowski, editors, *Intelligent Information Processing and Web Mining. Proceedings of the International IIS: IIPWM'06 Conference held in Ustron, Poland, June 19-22, 2006*, Advances in Soft Computing. Springer-Verlag, Berlin.

Daniel Zeman. 2001. How much will a RE-based preprocessor help a statistical parser? In *Proceedings of the Seventh International Workshop on Parsing Technologies (IWPT-2001), 17-19 October 2001, Beijing, China*. Tsinghua University Press.

Implementation of Croatian NERC System

Božo Bekavac
Department of Linguistics
University of Zagreb
Ivana Lučića 3, Zagreb, Croatia
bbekavac@ffzg.hr

Marko Tadić
Department of Linguistics
University of Zagreb
Ivana Lučića 3, Zagreb, Croatia
marko.tadic@ffzg.hr

Abstract

In this paper a system for Named Entity Recognition and Classification in Croatian language is described. The system is composed of the module for sentence segmentation, inflectional lexicon of common words, inflectional lexicon of names and regular local grammars for automatic recognition of numerical and temporal expressions. After the first step (sentence segmentation), the system attaches to each token its full morphosyntactic description and appropriate lemma and additional tags for potential categories for names without disambiguation. The third step (the core of the system) is the application of a set of rules for recognition and classification of named entities in already annotated texts. Rules based on described strategies (like internal and external evidence) are applied in cascade of transducers in defined order. Although there are other classification systems for NEs, the results of our system are annotated NEs which are following MUC-7 specification. System is applied on informative and noninformative texts and results are compared. F-measure of the system applied on informative texts yields over 90%.

1 Introduction

To produce a Named Entity Recognition and Classification (NERC) system for a lesser spread Slavic language like Croatian could be a task which differs a lot from the task of building such a system for a language like English, German or French. Compared to them, Croatian language has more elaborated inflectional system and more freedom in the word order within the sentence. Besides, the resources and tools needed for producing such a system (POS/MSD tagger, sentence segmentator, chunker, lexicons or gazetteers etc.) are not widely available.

But still we can say that even in languages with that kind of structural properties like Slavic languages have, named entities (NE) form a subset of natural language expressions that demonstrates relatively predictable structures. It could be questioned whether the relatively free word order in Croatian also covers the named entities (how much it influences their internal structure and their position in a sentence structure). What we also aim at in this paper is to investigate the possibility to describe NE with relatively simple rule-based systems i.e. whether it is possible to describe and classify NE in Croatian using regular grammars.

The next part of the paper describes basic methodology of our system. The third part presents strategies for NERC which have been converted into rules. The fourth part describes the architecture of the system while the fifth gives the results. The conclusion describes also possible future directions.

2 Methodology

This NERC system for Croatian is based on hand-made rules encoded in transducers which are applied in a cascade (Abney, 1996). The reason for selecting this method was simple. Since this is the first NERC system for Croatian, and there were no previous solutions for any particular NE class, we had to split the general NERC problem to a set of

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 11–18,
Prague, June 2007. ©2007 Association for Computational Linguistics

smaller locally manageable problems covering not just broad NE classes, but also their subclasses which were recognized by characteristic patterns. In such a way the set of rules could be kept under control and modules covering different parts of a problem could be called when needed in the runtime. In the same time the development time is shorter and the system is more consistent.

Every transducer in our system represents a local grammar (Gross, 1993) dedicated to the description of a part of a sentence i.e. local linguistic expression. The orientation to a local description where the simpler (and more certain) cases are solved first, followed by more complex ones, gives more precision to the whole system. This "island of certainty" principle (Abney, 1996:11) is also used in our NERC system.

The system uses the principle of the "longest match" as any other NERC system: in the case of more than one possible expression recognized by rules several different rules, system chooses the longest one. In this way potentially ambiguous NEs are being dynamically disambiguated (see e1 where *Madunić Ltd.* would be recognized and classified as organization NE because of the principle of the "longest match" which also included *Ltd* and thus avoiding matching only family name *Madunić* from the lexicon of names).

3 Strategies

In this section we will discuss the basic strategies that have been used in different NERC systems and their applicability for Croatian.

3.1 Internal and external evidence

The simple NER could be done by direct match of text with the list of NEs. Even if we previously solve the problem of inflection, such an approach would result with a lot of errors. In the example

> **e1:** Znali smo da je Madunić d.o.o. u vlasništvu njegova oca. (*We knew that Madunić Ltd is a property of his father.*)

the expression *Madunić* could be wrongly recognized and classified as a family name. Even that result can be questionable since it may happen that this very name is not in the list of family names. Better results could be gained by using more information i.e. features which already exist in NEs. One of features for personal names are titles such as *dr.*, *mr.*, *prof.*, *ing.* etc., for company names characteristic strings are *d.o.o.* (Ltd), *d.d.* (S.A., GmBH) etc. Such explicit strings are called *internal evidence* (McDonald 1996:22) and usually form a part of NE.

On the other hand the example such as:

> **e2:** Danas je stiglo pismo iz poduzeća "Đuro Đaković". (*A letter from the firm "Đuro Đaković" arrived today.*)

would yield simple person name *Đuro Đaković* if the contextual information of NE (i.e. string *poduzeća*/*the firm* and usage of quotes) is not taken into account. NEs often refer to certain classes such as institutions, hospitals, schools, persons, etc. Such contextual feature is called *external evidence* (McDonald 1996:22) and its recognition is mostly used as a classification criterion i.e. class membership proof. In the case illustrated by the following example:

> **e3:** U klinici za infektivne bolesti "Dr. Fran Mihaljević" tog je dana bila gužva. (*It was crowded that day at the clinic for infectuous diseases "Dr. Fran Mihaljević".*)

the external evidence is often decisive for NERC. In e3 the internal evidence (*Dr.*) represents a strong argument for a person NE, but only contextual external evidence (*klinici/the clinic* and quotes) gives the right solution.

The external evidences are crucial for NERC in any language but they also have an important role during the system development. They can be useful when a list of names is not complete – an external evidence is taking the role of an additional proof. They can also reduce the need for elaborated internal evidence checking when rules are being build.

The internal and external evidence is being used by all NERC systems such as LTG (Mikheev et al. 1999), FASTUS (Hobbs et al. 1997), Proteus (Yangarber, Grishman, 1998).

3.2 Dynamic lexicon

Sometimes during the processing there is a need for storing information which are relevant only for a current text/discourse/document. Such information are usually stored in a dynamic lexicon where temporarily relevant information are stored and used for the processing of a current document. Dynamic lexicon entries are being collected from the confident contexts and usually are being used for tagging words which could be NEs but there is not enough external evidences for that.

Dynamic lexicon could store all possible variants of a NE (a person) such as the full name and family name including middle initial, only family name, only name, only initials including all inflectional word-forms etc. In the case of companies, it could include the long company name, its shorter version and/or acronym. Distribution of acronyms shows that they frequently appear without internal and/or external evidences which are present with the full name (e.g. instead of the full name *Investicijsko-komercijalna banka*, in the text there is only *Banka* or only *IKB*). In such cases all tokens forming an NE and all their combinations are stored in the dynamic lexicon (Mikheev et al. 1999:5). In our case it would be also *Investicijsko-komercijalna, komercijalna banka, Investicijska banka*, and also an acronym derived from the first letters of all tokens (*IKB*).

Dynamic lexicon are used by a numer of NERC systems such as ones described in (Mikheev et al. 1998), (McDonald 1996) and (Piskorski et al. 2000).

3.3 Global word sequence checking

This strategy is used for solving complex ambiguities (Mikheev, 1999). The initial position in the sentence is one of such ambiguous spots. If the NE is complex e.g. has a conjuncted structure, its solving can be quite a difficult task. The following example from the newspaper can explain this:

> **e4:** Osiguranje Zagreb i Primošten potpisali su ugovor o suradnji. (*Insurance Zagreb and Primošten countersigned an agreement on cooperation.*)

> **e5:** Osiguranje Iviću i Horvatu nije isplatilo naknadu. (*Insurance didn't pay the benefit to Ivić and Horvat.*)

The token (*Osiguranje*) which in e4 is a part of NE (*Osiguranje Zagreb*) is also a common noun and is capitalized since it is in the initial sentence position. The second NE (*Primošten*) is from the list of locations but it could be also a part of conjunction (*Osiguranje Zagreb i Osiguranje Primošten*) which is shortened or forms a unique NE (*Osiguranje Zagreb i Primošten*). In the e5 there is no ambiguitiy since *Iviću* and *Horvatu* are person NEs and being in dative case clearly show that they do not belong to the same NE with *Osiguranje* (being in nominative case).

Conjunction *i* ('and') can be syntactically interpreted in two ways: it can serve as a connector of two separate NEs (*Pliva i INA*) or can be a part of NE (*Buhić i sinovi; Vodoopskrba i odvodnja*). This cases can be solved with a strategy that presupposes that at least there will be one unambiguous position for the same NE in the text. Solving the e4 example could be formulated in several steps. 1) all possible subsets of expression (*Osiguranje Zagreb i Primošten; Osiguranje Zagreb; Osiguranje Primošten; Zagreb i Primošten; Primošten…*) should be stored in a dynamic lexicon; 2) if any of this substrings is detected in the text in an unambiguous position:

> **e6:** Kapital Osiguranja Zagreb uvećan je tri puta. (*The capital of the Insurance Zagreb is enlarged three times.*)

> **e7:** Tvrtka Primošten d.d. izbjegla je stečaj. (*The firm Primošten d.d. avoided the bankrupcy.*)

the system can test that they are separate NEs and resolve the role of conjunction.

A proper solution for categorising *Primošten* is derived from this as well, since the coordinative conjunction *i* will usually connect the NEs from the same category (Mikheev, 1999).

This strategy is used in systems by Mikheev et al. (1999) and Wacholder (1997).

3.4 One sense per discourse

Ambiguous tokens, where the same string can refer to a common noun in common usage or as a part of NE, are quite common in texts (e.g. a token *Sunce* in initial sentence position can be a common noun but it has been recorded that it can also be a name of investment fund or insurance company).

Since texts are meant to be understood by readers (even when shortening and compressing procedures are used by authors) it is very rare that the same token has different meanings within the same text. Gale, Church and Yarowsky (1992) formed a hypothesis that ambiguous words have a strong tendency of keeping a single meaning in the same text/discourse. It has been experimentally proven up to 98% of cases. Therefore, detecting at least one unambiguous position for an ambiguous word enables the system to successfully solve all other ambiguous positions for this word.

3.5 Filtering of the false candidates

Specific type of problem for NERC systems pose expressions which have a structure similar to NE, but are not NEs:

> **e8:** `Pripreme za Atenu 2004 približavaju se završetku.` (*Preparations for the Athens 2004 are coming to the end.*)
>
> **e9:** `Pogled nam se pružao na cijelu Atenu.` (*A view to the whole Athens was in front of us.*)

In e8 string *Atenu 2004* refers to the Olympic games held in Athens 2004 and not to location NE. According to MUC specification, this should not be marked as NE. In e9 *Atenu* refers to location and should be marked as NE.

There are two possible solutions for elimination of this cases: 1) a context should be expressive enough that it can be covered by a special rule; 2) a list of false NE candidates i.e. NE-like expressions which have to be eliminated from the further processing.

It is better to discard the false NE candidates at the beginning (Karkaletis et al 1999:130) because it reduces the need for further processing and testing. The false NE candidates should not have to be deleted from the text, a better solution is to mark them with a special tag which will be deleted just before output but in the same time it will signal to the system to avoid the processing of that part of text.

Processing of false NE candidates is described thoroughly in (Stevenson, Gaizauskas, 1999:293).

4 Architecture of the system

For developing, testing and applying our NERC system we were using Intex, a well known development environment for making formal descriptions of natural languages using FSTs and their immediate application on large corpora in real-time (Silberztein 2000:8).

Our system was designed to allow the modular processing of Croatian on three levels: 1) token (single-word units) segmentation; 2) sentence segmentation; 3) multi-word units (collocations, syntagms). These modules were designed for this system but they can be used individually in any other system for processing Croatian.

Lists of personal and family names are also important for this system. We were using a list of 15,000 male and female personal names accompanied by 56,000 family names registered in the Republic of Croatia (Boras; Mikelić; Lauc 2003:224). This list was expanded to a full word-form list for every name according to the MulTextEast specification for lexica (Erjavec et al. 2003).

The rules were manually developed and tested on a subcorpus of Croatian National Corpus (Tadić, 2002) which size was 60 million of tokens of newspaper texts. The rules were coded as Finite State Transducers using Intex's graphical interface.

The system (see figure 1) consists of several sequenced modules which are applied after the tokenizaton and sentence segmentation:

1. Lexical processing: application of lexicons of common words and proper names. Unrecognized tokens are further processed with transducers which are based on characteristic endings for MSD categorization.

2. Rules (phase 1) which have the highest certainty i.e. process unambiguous text segments are being applied after the preprocessing stage. In this manner a large part of all NEs is being detected thus giving the firm anchors for the rules (phase 2);

3. Lexicon filtering: some lexical entries are highly ambiguous and make application of relaxed rules even more complex (e.g. *Kina* in Croatian can be a common noun and location NE as well. Filtering such highly frequent and ambiguous common words significantly increases results in the second phase.

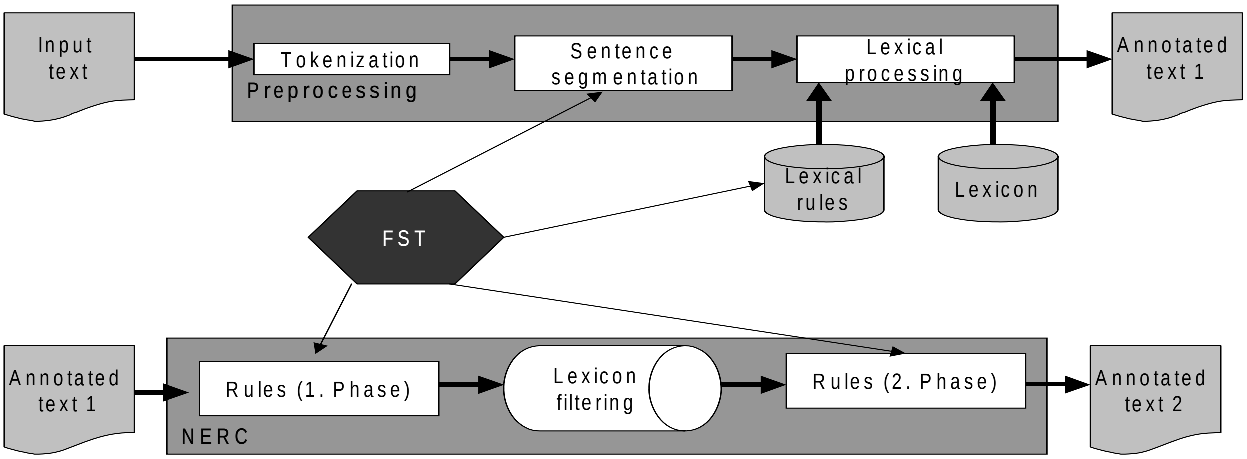


Figure 1. The general architecture of the system.

4. Rules (phase 2): all unrecognized NEs in phase 1 (mostly because of lack of supportive co-text information) are processed with new rules which are relaxed. Constraints are relaxed, but thanks to filtered lexicon precision are still rather high.

Since the overall number of rules is 106 and the description of their precise ordering and mutual interdependence would surpass the limitations of this article, we would like to exemplify the general format of the rules with the rule for detecting person NEs which include external evidence such as function of that person. Since functions can appear before or after the person NE, this rule has been stored as a separate local grammar which is being called as needed.

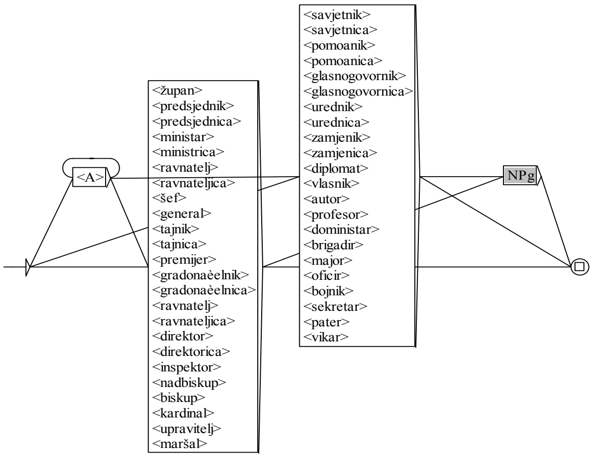


Figure 2. Graph for functions (funkcije.grf).

Beside the function name, an attribute <A> can appear on the left and NP in genitive case [NPg] can appear on the right of function name.

This local grammar ([funkcije] in grey) is being called in cascade from two other grammars for person NE detection such as:

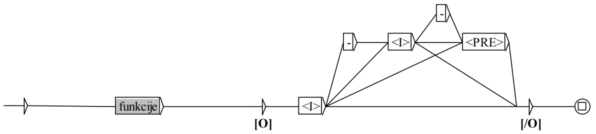


Figure 3. Graph for functions + names.

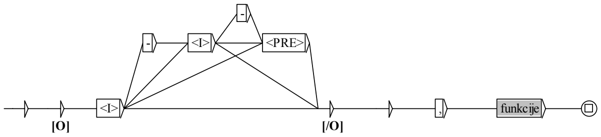
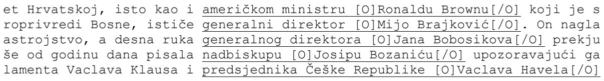


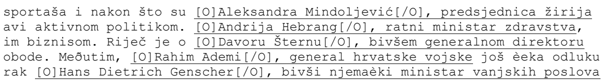
Figure 4. Graph for names + functions.

In figures 3 and 4 <I> represents a personal name recognized from the list of personal names while <PRE> represents a capitalized token. [O] and [/O] are tags that system inserts for person NE annotation. In this way potentially ambiguous NEs like *Predsjednik Microsofta* and *Predsjednik Šeks* could be resolved since only *Šeks* belongs to a list of personal names. The grammar in figure 3 can recognize cases such as:



```
et Hrvatskoj, isto kao i američkom ministru [O]Ronaldu Brownu[/O] koji je s
roprivredi Bosne, ističe generalni direktor [O]Mijo Brajković[/O]. On nagla
astrojstvo, a desna ruka generalnog direktora [O]Jana Bobosikova[/O] prekju
še od godinu dana pisala nadbiskupu [O]Josipu Bozaniću[/O] upozoravajući ga
lamenta Vaclava Klausa i predsjednika Češke Republike [O]Vaclava Havela[/O]
```

while the grammar in figure 4 can recognize cases such as:



```
sportaša i nakon što su [O]Aleksandra Mindoljević[/O], predsjednica žirija
avi aktivnom politikom. [O]Andrija Hebrang[/O], ratni ministar zdravstva,
im biznisom. Riječ je o [O]Davoru Šternu[/O], bivšem generalnom direktoru
obode. Međutim, [O]Rahim Ademi[/O], general hrvatske vojske još èeka odluku
rak [O]Hans Dietrich Genscher[/O], bivši njemaèki ministar vanjskih poslova
```

All local grammars for detecting personal NEs are being called from a grammar on upper level:

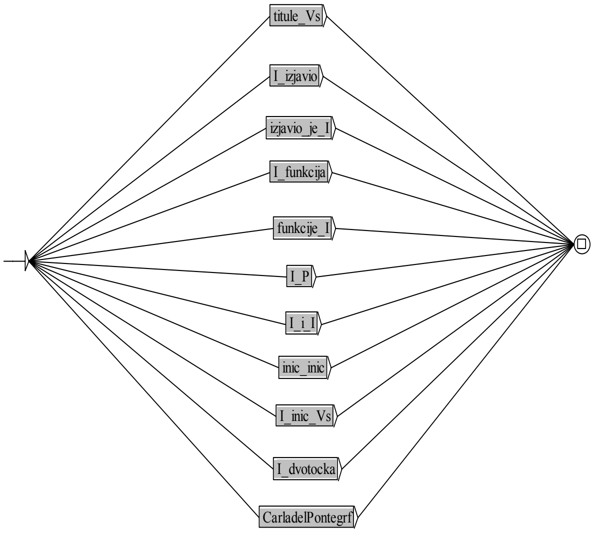


Figure 5: Graph with all person NE graphs

Similar set of rules and modular local grammars has been developed for other NE categories.

The order of applying rules (i.e. local grammars) plays important role in our NERC system. There are at least two reasons for that.

1) Certain rule can be valid for a NE which can be part of a larger NE. Rules for organization NE detection should be applied prior to rules for person NE detection. In this way correct categorisation is being achieved (e.g. *Đuro Đaković holding d.d.* where a person NE should not be used and subsumed under larger organization NE). Even if both grammars are applied simultaneously, still the principle of "longest match" would yield the correct categorisation (Poibeau, 2000). The same ordering should be kept in mind for other types of NEs which could be subsumed (e.g. dates or locations within the names of streets etc.).

2) The degree of certainty is decisive for rule ordering: the most certain NEs are being processed at the beginning and thus lowers the ambiguity also within the same category.

5 Results and discussion

Our NERC system for Croatian was tested on two types of texts: newspaper articles from *Večernji list* (economy and internal affairs, 350 articles from 2005-01, 137.547 tokens) and two textbooks from the history of arts and culture (143.919 tokens) (Maković, 1997; Žmegač, 1998). The results for newspaper texts are given in Table 1, while results for textbooks are given in Table 2.

F-measure of the whole system calculated as average from F-measures of all categories is 0.92. Since all NE categories are not equally represented in texts, more realistic measure of system efficiency can be acquired by counting all NEs that current version of a system with this set of rules should detect and categorize in a text. In this case F-measure drops to 0.90 which is still very good result.

The same rules applied to another genre (textbooks) show a significant drop in the accuracy of the system. Precision is still at 0.79 but recall is at 0.47 thus resulting with F-measure at 0.59. The most serious drop is in personal and location names. Possible explanation could be that in textbooks used for testing there is a lot of unknown, possibly foreign, names but this has to be checked in detail on more different genres.

Compared to a similar system for NERC in French texts (Poibeau; Kosseim 2001:148), where also Intex was used as a development environment, we got similar results. System developed for French yielded 0.9 for informative texts and 0.5 in noninformative texts (prose).

The example of the input and output from our system can be seen at http://hnk.ffzg.hr/nerc/.

Theoretically syntactic rules in Croatian do allow central embedding in NPs thus splitting them in two separate strings. If we apply this rule to a NERC domain, we could think of a construction which consist of function and personal name:

> **e10:** **bivši hrvatski predsjednik*, koji je stvorio hrvatsku državu, *Franjo Tuđman*...(**former Croatian president, who founded a Croatian state, Franjo Tuđman*...)

	Person	Organization	Location	Percentage	Currency	Time
Precision	0.95	0.93	0.98	0.99	0.99	0.94
Recall	0.69	0.86	0.93	0.99	0.99	0.90
F-measure	0.79	0.89	0.95	0.99	0.99	0.92

Table 1: Results for newspaper articles

	Person	Organization	Location	Percentage	Currency	Time
Precision	0.65	0.69	0.61	0.95	0.92	0.91
Recall	0.35	0.38	0.31	0.66	0.61	0.53
F-measure	0.46?	0.49	0.41	0.78	0.73	0.67

Table 2: Results for textbooks

In practice constructions of this type were never detected even in a very large corpus (>100 Mw). This led us to a conclusion that in spite the relatively free word order in Croatian, for NERC systems regular grammars could be sufficient instead of stronger formalism such as context-free grammars. NEs are local phenomena in sentences and are usually kept in one constituent. It looks like the free word order allows recombination of constituents (scrambling) while withing the constituents it is not allowed and they could be locally recognized by regular grammars. Although context-free grammars encompass regular ones, the development time for regular grammars, particularly if they are built as small-scale local grammars which are cascaded later, is much shorter and developers have stronger control over of each module, its input and output.

6 Future directions

Although it features in some areas quite promising results, this system if far from being complete. Our future directions could be: 1) testing the system on a whole different range of genres with possible rule adaptation for each genre; 2) widening the list of person and family names to include foreign names; 3) thorough analysis and typology of most typical errors; 4) include also other NEs classification schemes which go beyond MUC-7 specification; 5) since this system highly depends on Intex runtime library under which it has been designed, it is not possible to distribute it as a stand-alone application. We would like to reprogram the whole set of rules on a different platform or programming language. In this way this system can became a core of a web-based service for NERC in Croatian which is also one of our intentions.

Acknowledgments

This work was partially supported by the Ministry of the Science, Education and Sports of the Republic of Croatia within the project 130-1300646-0645 and partially by Flemish and Croatian governments within the joint CADIAL project.

References

Abney, Steven. 1996. *Partial Parsing via Finite-State Cascades*, Journal of Natural Language Engineering 2 (4):337–344.

Damir Boras, Nives Mikelić, Davor Lauc. 2003. *Leksička flektivna baza podataka hrvatskih imena i prezimena*, Modeli znanja i obrada prirodnog jezika – Zbornik radova, Radovi Zavoda za informacijske studije (vol. 12):219–237.

Tomaž Erjavec (ed.). 2001. *Specifications and Notations for MULTEXT-East Lexicon Encoding. Edition Multext-East/Concede Edition*, March, 21, p. Available at [http://nl.ijs.si/ME/ V2/msd/html/].

Friburger, Nathalie; Maurel, Denis. 2004. *Finite-state transducer cascades to extract named entities in texts,* Theoretical Computer Science, 313(1):93–104.

William Gale, Kenneth Church, David Yarowsky. 1992. *One Sense per Discourse*, Proceedings of the 4th DARPA Speech and Natural Language Workshop, Harriman, NY:233–237.

Maurice Gross. 1993. *Local grammars and their representation by finite automata*, Data Description, Discourse (ed. M. Hoey), Harper-Collins, London:26–38.

Jerry R Hobbs, Douglas E. Appelt, John Bear, David Israel, Megumi Kameyama, Mark Stickel, Mabry Tyson. 1997. *FASTUS: A cascaded finite-state transducer for extracting information from natural language text*, Finite State Devices for Natural Language Processing, (ed. Roche, E.; Schabes, Y.), MIT Press, Cambridge, MA:383–406.

Vangelis Karkaletsis, Georgios Paliouras, Georgios Petasis, Natasa Manousopoulou, Constantine D. Spyropoulos. 1999. *Named-Entity Recognition from Greek and English Texts*, Journal of Intelligent and Robotic Systems, 26(2):123–135.

Maković, Zvonko. 1997. *Vilko Gecan*, Matica hrvatska, Zagreb.

David McDonald. 1996. *Internal and external evidence in the identification and semantic categorization of proper names*, Corpus Processing for Lexical Acquisition, chapter 2, ed. Boguraev; Pustejovsky, The MIT Press, Cambridge, MA:21–39.

Andrei Mikheev, Claire Grover,Marc Moens. 1998. *Description of the LTG system used for MUC-7*, Proceedings of the 7th Message Understanding Conference (MUC-7), Fairfax, Virginia

Andrei Mikheev, Claire Grover,Marc Moens. 1999. *Named Entity Recognition without Gazetteers*, Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics, Bergen:1–8.

Andrei Mikheev. 1999. *A Knowledge-free Method for Capitalized Word Disambiguation*, Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics:159–166.

Jakub Piskorski, Günter Neumann. 2000. *An Intelligent Text Extraction and Navigation System*, Proceedings of the 6th International Conference on Computer-Assisted Information Retrieval (RIAO'00), Paris

Thierry Poibeau. 2000. *A Corpus-based Approach to Information Extraction*, Journal of Applied Systems Studies, 1(2):254–267.

Thierry Poibeau, Leila Kosseim. 2001. *Proper Name Extraction from Non-Journalistic Texts*, Computational Linguistics in the Netherlands 2000: Selected Papers from the Eleventh CLIN Meeting, W. Daelemans, K. Sima'an, J. Veenstra, J. Zavrel (ed.), Rodopi, Amsterdam:144–157.

Max Silberztein. 1999. *INTEX: a Finite State Transducer toolbox,* Theoretical Computer Science #231:1, Elsevier Science

Max Silberztein. 2000. *INTEX Manual*. ASSTRIL, Paris

Mark Stevenson, Robert Gaizauskas. 1999. *Using Corpus-derived Name Lists for Named Entity Recognition*, Proceedings of the sixth conference on Applied natural language processing, Seattle, Washington, Morgan Kaufmann Publishers Inc.:290–295.

Marko Tadić. 2002. *Building the Croatian National Corpus*. LREC2002 Proceedings, Las Palmas, ELRA, Pariz-Las Palmas, Vol. II:441-446.

Nina Wacholder, Yael Ravin, Misook Choi. 1997. *Disambiguation of Proper Names in Text*, Proceedings of the Fifth Conference on Applied Natural Language Processing:202–208.

Roman Yangarber, Ralph Grishman. 1998. *NYU: Description of the Proteus/PET system as used for MUC-7 ST*, Proceedings of the 7th Message Understanding Conference (MUC-7), Fairfax, Virginia.

Žmegač, Viktor. 1998. *Bečka moderna*, Matica hrvatska, Zagreb.

A Language Independent Approach for Name Categorization and Discrimination

Zornitsa Kozareva
Departamento de Lenguajes
y Sistemas Informáticos
Universidad de Alicante
Alicante, Spain
zkozareva@dlsi.ua.es

Sonia Vázquez
Departamento de Lenguajes
y Sistemas Informáticos
Universidad de Alicante
Alicante, Spain
svazquez@dlsi.ua.es

Andrés Montoyo
Departamento de Lenguajes
y Sistemas Informáticos
Universidad de Alicante
Alicante, Spain
montoyo@dlsi.ua.es

Abstract

We present a language independent approach for fine-grained categorization and discrimination of names on the basis of text semantic similarity information. The experiments are conducted for languages from the Romance (Spanish) and Slavonic (Bulgarian) language groups. Despite the fact that these languages have specific characteristics as word-order and grammar, the obtained results are encouraging and show that our name entity method is scalable not only to different categories, but also to different languages. In an exhaustive experimental evaluation, we have demonstrated that our approach yields better results compared to a baseline system.

1 Introduction

1.1 Background

Named Entity (NE) recognition concerns the detection and classification of names into a set of categories. Presently, most of the successful NE approaches employ machine learning techniques and handle simply the person, organization, location and miscellaneous categories. However, the need of the current Natural Language Applications impedes specialized NE extractors which can help for instance an information retrieval system to determine that a query about "Jim Henriques guitars" is related to the person "Jim Henriques" with the semantic category musician, and not "Jim Henriques" the composer. Such classification can aid the system to rank or return relevant answers in a more accurate and appropriate way.

So far, the state-of-art NE recognizers identify that "Jim Henriques" is a person, but do not subcategorize it. There are numerous drawbacks related to the fine-grained NE issue. First, the systems need hand annotated data which are not available for multiple categories, because their creation is time-consuming, requires supervision by experts, a predefined fine-grained hierarchical structure or ontology. Second, there is a significant lack of freely available or developed resources for languages other than English, and especially for the Eastern European ones.

The World Wide Web is a vast, multilingual source of unstructured information which we consult daily in our native language to understand what the weather in our city is or how our favourite soccer team performed. Therefore, the need of multilingual and specialized NE extractors remains and we have to focus on the development of language independent approaches.

Together with the specialized NE categorization, we face the problem of name ambiguity which is related to queries for different people, locations or companies that share the same name. For instance, Cambridge is a city in the United Kingdom, but also in the United States of America. ACL refers to "The Association of Computational Linguistics", "The Association of Christian Librarians" or to the "Automotive Components Limited". Googling the name "Boyan Bonev" returns thousands of documents where some are related to a member of a robot vision group in Alicante, a teacher at the School

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 19–26,
Prague, June 2007. ©2007 Association for Computational Linguistics

of Biomedical Science, a Bulgarian schoolboy that participated in computer science competition among others. So far, we have to open the documents one by one, skim the text and decide to which "Boyan Bonev" the documents are related to. However, if we resolve the name disambiguation issue, this can lead to an automatic clustering of web pages talking about the same individual, location or ogranization.

1.2 Related Work

Previously, (Pedersen et al., 2005) tackled the name discrimination task by developing a language independent approach based on the context in which the ambiguous name occurred. They construct second order co-occurrence features according to which the entities are clustered and associated to different underlying names. The performance of this method ranges from 51% to 73% depending on the pair of named entities that have to be disambiguated. Similar approach was developed by (Bagga and Baldwin, 1998), who created first order context vectors that represent the instance in which the ambiguous name occurs. Their approach is evaluated on 35 different mentions of John Smith, and the f-score is 84%.

For fine-grained person NE categorization, (Fleischman and Hovy, 2002) carried out a supervised learning for which they deduced features from the local context in which the entity resides, as well as semantic information derived from the topic signatures and WordNet. According to their results, to improve the 70% coverage for person name categorization, more sophisticated features are needed, together with a more solid data generation procedure. (Tanev and Magnini, 2006) classified geographic location and person names into several subclasses. They use syntactic information and observed how often a syntactic pattern co-occurs with certain member of a given class. Their method reaches 65% accuracy. (Pasca, 2004) presented a lightly supervised lexico-syntactic method for named entity categorization which reaches 76% when evaluated with unstructured text of Web documents.

(Mann, 2002) populated a fine-grained proper noun ontology using common noun patterns and following the hierarchy of WordNet. They studied the influence of the newly generated person ontology in a Question Answering system. According to the obtained results, the precision of the ontology is high, but still suffers in coverage. A similar approach for the population of the CyC Knowledge Base (KB) was presented in (Shah et al., 2006). They used information from the Web and other electronically available text corpora to gather facts about particular named entities, to validate and finally to add them to the CyC KB.

In this paper, we present a new text semantic similarity approach for fine-grained person name categorization and discrimination which is similar to those of (Pedersen et al., 2005) and (Bagga and Baldwin, 1998), but instead of simple word co-occurrences, we consider the whole text segment and relate the deduced semantic information of Latent Semantic Analysis (LSA) to trace the text cohesion between thousands of sentences containing named entities which belong to different fine-grained categories or individuals. Our method is based on the word sense discrimination hypothesis of Miller and Charles (1991) according to which words with similar meaning are used in similar context, hence in our approach we assume that the same person or the same fine-grained person category appears in the similar context.

2 NE categorization and discrimination with Latent Semantic Analysis

LSA has been applied successfully in many areas of Natural Language Processing such as Information Retrieval (Deerwester et al., 1990), Information Filtering (Dumais, 1995) , Word Sense Disambiguation (Shütze, 1998) among others. This is possible because LSA is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in discourse. It uses no humanly constructed dictionaries or knowledge bases, semantic networks, syntactic or morphological analyzers, because it takes only as input raw text which is parsed into words and is separated into meaningful passages. On the basis of this information, LSA extracts a list of semantically related word pairs or rank documents related to the same topic.

LSA represents explicitly terms and documents in a rich, highly dimensional space, allowing the underlying "latent", semantic relationships between terms and documents to be exploited. LSA relies on the constituent terms of a document to suggest the document's semantic content. However, the LSA model views the terms in a document as somewhat unreliable indicators of the concepts contained in the document. It assumes that the variability of word choice partially obscures the semantic structure of the document. By reducing the original dimensionality of the term-document space with Singular Value Decomposition to a matrix of 300 columns, the underlying, semantic relationships between documents are revealed, and much of the "noise" (differences in word usage, terms that do not help distinguish documents, etc.) is eliminated. LSA statistically analyzes the patterns of word usage across the entire document collection, placing documents with similar word usage patterns near to each other in the term-document space, and allowing semantically-related documents to be closer even though they may not share terms.

Taking into consideration these properties of LSA, we thought that instead of constructing the traditional term-document matrix, we can construct a term-sentence matrix with which we can find a set of sentences that are semantically related and talk about the same person. The rows of the term-sentence matrix correspond to the words of the sentence where the NE has to be categorized or discriminated (we call this sentence target sentence), while the columns correspond to the rest of the sentences with NEs. The cells of the matrix show the number of times a given word from the target sentence co-occurs in the rest of the sentences. When two columns of the term-sentence matrix are similar, this means that the two sentences contain similar words and are therefore likely to be semantically related. When two rows are similar, then the corresponding words occur in most of the same sentences and are likely to be semantically related.

In this way, we can obtain semantic evidence about the words which characterize a given person. For instance, a *football player* is related to words as *ball*, *match*, *soccer*, *goal*, and is seen in phrases such as "X *scores a goal*", "*Y is penalized*". Meanwhile, a *surgeon* is related to words as *hospital*, *patient*, *operation*, *surgery* and is seen in phrases such as "*X operates Y*", "*X transplants*". Evidently, the category football player can be distinguished easily from that of the surgeon, because both person names occur and relate semantically to different words.

Another advantage of LSA is its property of language independence, and the ability to link several flexions or declanations of the same term. This is especially useful for the balto-slavonic languages which have rich morphology. Once the term-sentence approach is developed, practically there is no restrain for LSA to be applied and extended to other languages. As our research focuses not only on the resolution of the NE categorization and discrimination problems as a whole, but also on the language independence issue, we considered the LSA's usage are very appropriate.

3 Development Data Set

For the development of our name discrimination and classification approach, we used the Spanish language. The corpora we worked with is the EFE94-95 Spanish news corpora, which were previously used in the CLEF competitions[1]. In order to identify the named entities in the corpora, we used a machine learning based named entity recognizer (Kozareva et al., 2007).

For the NE categorization and discrimination experiments, we used six different named entities, for which we assumed a-priory to belong to one of the two fine-grained NE categories PERSON_SINGER and PERSON_PRESIDENT. The president names are Bill Clinton, George Bush and Fidel Castro, and the singer names are Madonna, Julio Iglesias and Enrique Iglesias. We have selected these names for our experiment, because of their high frequency in the corpora and low level of ambiguity.

Once we have selected the names, we have collected a context of 10, 25, 50 and 100 words from the left and from the right of the NEs. This is done in order to study the influence of the context for the NE discrimination and categorization tasks, and especially how the context window affects LSA's performance. We should note that the context for the NEs is obtained from the text situated between the text tags. During the creation of the context window, we used only the words that belong to the document in which the NE is detected. This restriction is imposed, because if we use words from previous or following documents, this can influence and change

[1] http://www.clef-campaign.org/

the domain and the topic in which the NE is seen. Therefore, NE examples for which the number of context words does not correspond to 10, 25, 50 or 100 are directly discarded.

From the compiled data, we have randomly selected different NE examples and we have created two data sets: one with 100 and another with 200 examples per NE. In the fine-grained classification, we have substituted the occurrence of the president and singer names with the obfuscated form President_Singer. While for the NE discrimination task, we have replaced the names with the M_EI_JI_BC_GB_FC label. The first label indicates that a given sentence can belong to the president or to the singer category, while the second label indicates that behind it can stand one of the six named entities. The NE categorization and discrimination experiments are carried out in a completely unsupervised way, meaning that we did not use the correct name and name category until evaluation.

4 Experimental Evaluation

4.1 Experimental Settings

As mentioned in Section 2, to establish the semantic similarity relation between a sentence with an obfuscated name and the rest of the sentences, we use LSA[2]. The output of LSA is a list of sentences that best matches the target sentence (e.g. the sentence with the name that has to be classified or discriminated) ordered by their semantic similarity score. Strongly similar sentences have values close to 1, and dissimilar sentences have values close to 0.

In order to group the most semantically similar sentences which we expect to refer to the same person or the same fine-grained category, we apply the graph-based clustering algorithm PoBOC (Cleuziou et al., 2004). We construct a new quadratic sentence-sentence similarity matrix where the rows stand for the sentence we want to classify, the columns stand for the sentences in the whole corpus and the values of the cells represent the semantic similarity scores derived from LSA.

On the basis of this information, PoBOC forms two clusters whose performance is evaluated in terms of precision, recall, f-score and accuracy which can be derived from Table 1.

number of	Correct PRESIDENT	Correct SINGER
Assigned PRESIDENT	a	b
Assigned SINGER	c	d

Table 1: Contingency table

We have used the same experimental setting for the name categorization and discrimination problems.

4.2 Spanish name categorization

In Table 2, we show the results for the Spanish fine-grained categorization. The detailed results are for the context window of 50 words with 100 and 200 examples. All runs, outperform a simple baseline system which returns for half of the examples the fine-grained category PRESIDENT and for the rest SINGER. This 50% baseline performance is due to the balanced corpus we have created. In the column *diff.*, we show the difference between the 50% baseline and the f-score of the category. As can be seen the f-scores reaches 90%, which is with 40% more than the baseline. According to the z' statistics with confidence level of 0.975, the improvement over the baseline is statistically significant.

SPANISH						
cont/ex	Category	P.	R.	A.	F.	diff.
50/100	PRESIDENT	90.38	87.67	88.83	89.00	+39.00
	SINGER	87.94	90.00	88.33	88.96	
50/200	PRESIDENT	90.10	94.33	91.92	92.18	+42.00
	SINGER	94.04	89.50	91.91	91.71	

Table 2: Spanish NE categorization

During the error analysis, we found out that the PERSON_PRESIDENT and PERSON_SINGER categories are distinguishable and separable because of the well-established semantic similarity relation among the words with which the NE occurs.

A pair of president sentences has lots of strongly related words such as *president:meeting*, *president:governmen*t, which indicates high text cohesion, while the majority of words in a president–singer pair are weakly related, for instance *president:famous*, *president:concert*. But still we found out ambiguous pairs such as *president:company*, where the president relates to a president of a country, while the company refers to a musical enter-

[2] http://infomap-nlp.sourceforge.net/

name	c10	c25	c50	c100
Madonna	**63.63**	**61.61**	63.16	**79.45**
Julio Iglesias	58.96	56.68	66.00	79.19
Enrique Iglesias	**77.27**	**80.17**	**84.36**	**90.54**
Bill Clinton	52.72	48.81	**74.74**	73.91
George Bush	49.45	41.38	60.20	67.90
Fidel Castro	**61.20**	**62.44**	**77.08**	**82.41**

Table 3: Spanish NE discrimination

prize. Such information confuses LSA's categorization process and decreases the NE categorization performance.

4.3 Spanish name discrimination

In a continuation, we present in Table 3 the f-scores for the Spanish NE discrimination task with the 10, 25, 50 and 100 context windows. The results show that the semantic similarity method we employ is very reliable and suitable not only for the NE categorization, but also for the NE discrimination. A baseline which always returns one and the same person name during the NE discrimination task is 17%. From the table can be seen that all names outperform this baseline. The f-score performance per individual name ranges from 42% to 90%. The results are very good, as the conflated names (three presidents and three singers) can be easily obfuscated, because they share the same domain and occur with the same semantically related words.

The three best discriminated names are Enrique Iglesias, Fidel Castro and Madonna. The name Fidel Castro is easily discriminated due to its characterizing words *Cuba*, *CIA*, *Cuban president*, *revolution*, *tyrant*. All sentences having these words or synonyms related to them are associated to Fidel Castro.

Bill Clinton occurred many times with the words *democracy*, *Boris Yeltsin*, *Halifax*, *Chelsea* (the daughter of Bill Clinton), *White House*, while George Bush appeared with *republican*, *Ronald Reigan*, *Pentagon*, *war in Vietnam*, *Barbara Bush* (the wife of George Bush).

During the data compilation process, the examples for Enrique Iglesias are considered to belong to the Spanish singer. However, in reality some examples of Enrique Iglesias talked about the president of a financial company in Uruguay or political issues. Therefore, this name was confused with Bill Clinton, because they shared semantically related words such as *bank*, *general secretary*, *meeting*, *decision*, *appointment*.

The discrimination process for the singer names is good, though Madonna and Julio Iglesias appeared in the context of *concerts*, *famous*, *artist*, *magazine*, *scene*, *backstage*. The characterizing words for Julio Iglesias are *Chabeli* (the daughter of Julio Iglesias), *Spanish*, *Madrid*, *Iberoamerican*. The name Madonna occurred with words related to a picture of Madonna, a statue in a church of Madonna, the movie Evita.

Looking at the effect of the context window for the NE discrimination task, it can be seen that the best performances of 90% for Enrique Iglesias, 82% for Fidel Castro and 79% for Madonna are achieved with 100 words from the left and from the right of the NE. This shows that the larger context has better discrimination power.

4.4 Discussion

After the error analysis, we saw that the performance of our approach depends on the quality of the data source we worked with. Although, we have selected names with low degree of ambiguity, during the data compilation process for which we assumed that they refer 100% to the `SINGER` or `PRESIDENT` categories, during the experiments we found out that one and the same name can refer to three different individuals. This was the case of Madonna and Enrique Iglesias. From one side this impeded the fine-grained categorization and discrimination processes, but opened a new line for research.

In conclusion, the conducted experiments revealed a series of important observations. The first one is that the LSA's term-sentence approach performs better with a higher number of examples, because they provide more semantic information. In addition to the number of examples, the experiments show that the influence of the context window for the name discrimination is significant. The discrimination power is better for larger context windows and this is also related to the expressiveness of the language.

Second, our name categorization and discrimination approach outperforms the baseline with 30%. Finally, LSA is a very appropriate approximation for the resolution of the NE categorization and discrimination tasks. LSA also gives logical explanation about the classification decision of the person names, providing a set of words characterizing the category or simply a list of words describing the individual we want to classify.

5 Adaptation to Bulgarian

5.1 Motivation

So far, we have discussed and described the development and the performance of our approach with the Spanish language. The obtained results and observations, serve as a base for the context extraction and the experimental setup for the rest of the languages which we want to study. However, to verify the multilingual performance of the approach, we decided to carry out an experiment with a language which is very different from the Romance family.

For this reason, we choose the Bulgarian language, which is the earliest written Slavic language. It dates back from the creation of the old Bulgarian alphabet Glagolista, which was later replaced by the Cyrillic alphabet. The most typical characteristics of the Bulgarian language are the elimination of noun declension, suffixed definite article, lack of a verb infinitive and complicated verb system.

The Bulgarian name discrimination data is extracted from the news corpus Sega2002. This corpus is originally prepared and used in the CLEF competitions. The corpus consists of news articles organized in different XML files depending on the year, month, and day of the publication of the news. We merged all files into a single one, and considered only the text between the text tags. In order to ease the text processing and to avoid encoding problems, we transliterated the Cyrillic characters into Latin ones.

The discrimination data in this experiment consists of the city, country, party, river and mountain categories. We were interested in studying not only the multilingual issue of our approach, but also how scalable it is with other categories. The majority of the categories are locations and only one corresponds to organization. In Table 4, we shows the number of names which we extracted for each one of the categories.

5.2 Bulgarian data

The cities include the capital of Bulgaria – Sofia, the second and third biggest Bulgarian cities – Plovdiv and Varna, a city from the southern parts of Bulgaria – Haskovo, the capital of England – London and the capital of Russia – Moskva. The occurrences of these examples are conflated in the ambiguous name CITY.

For countries we choose Russia (Rusiya)[3], Germany (Germaniya), France (Franciya), Turkey (Turciya) and England (Angliya). The five names are conflated into COUNTRY.

The organizations we worked with are the two leading Bulgarian political parties. BSP (Balgarska Socialisticeska Partija, or Bulgarian Socialist Party) is the left leaning party and the successor to the Bulgarian Communist Party. SDS (Sayuz na demokratichnite sili, or The Union of Democratic Forces) is the right leaning political party. The two organizations are conflated into PARTY.

For the RIVER category we choose Danube (Dunav) which is the second longest river in Europe and passes by Bulgaria, Maritsa which is the longest river that runs solely in the interior of the Balkans, Struma and Mesta which run in Bulgaria and Greece.

The final category consists of the oldest Bulgarian mountain situated in the southern part of Bulgaria – Rhodope (Rodopi), Rila which is the highest mountain in Bulgaria and on the whole Balkan Peninsula, and Pirin which is the second highest Bulgarian mountain after Rila. The three mountain names are conflated and substituted with the label MOUNTAIN.

5.3 Bulgarian name discrimination

The experimental settings coincide with those presented in Section 4 and the obtained results are shown in Table 4. The performance of our approach ranges from 32 to 81%. For the five categories, the best performance is achieved for those names that have the majority number of examples.

For instance, for the CITY category, the best performance of 79% is reached with Sofia. TAs we have previously mentioned, this is due to the fact that LSA has more evidence about the context in which Sofia appears. It is interesting to note that the city

[3] this is the Bulgarian transliteration for Russia

Category	Instance	Total	P	R	F
City	Plovdiv	1822	44.42	83.87	58.08
	Sofiya	**5633**	71.39	89.79	**79.54**
	Varna	1042	32.02	82.64	46.17
	Haskovo	140	21.09	69.29	32.33
	London	751	31.32	84.82	45.74
	Moskva	1087	39.47	88.22	54.53
Country	**Rusiya**	**2043**	55.83	86.19	**67.77**
	Germaniya	1588	40.72	77.96	53.50
	Francia	1352	37.27	77.81	50.39
	Turciya	1162	43.23	84.08	57.10
	Angliya	655	29.67	72.67	42.14
Party	BSP	2323	42.54	99.35	59.57
	SDS	**3916**	64.86	98.85	**78.32**
River	**Dunav**	**403**	85.39	76.92	**80.94**
	Marica	203	77.88	83.25	80.47
	Mesta	81	63.64	95.06	76.24
	Struma	37	56.67	91.89	70.10
Mountain	Rila	101	70.22	91.09	79.31
	Pirin	294	75.11	57.48	65.12
	Rodopi	**135**	71.04	96.29	**81.76**

Table 4: Bulgarian NE discrimination

Varna forms part of weak named entities such as the University of Varna, the Major house of Varna. Although, this strong entity is embedded into the weak ones, practically Varna changes its semantic category from a city into university, major house. This creates additional ambiguity in our already conflated and ambiguous names. In order to improve the performance, we need a better data generation process where the mixture of weak and strong entities will be avoided.

The same effect of best classification for majority sense is observed with the COUNTRY category. The best performance of 67% is obtained for Russia. The other country which is distinguished significantly well is Turkey. The 57% performance is from 5 to 10% higher compared to the performances of Germany, England and France. This is due to the context in which the names occur. Turkey is related to trading with Bulgaria and emigration, meanwhile the other countries appear in the context of the European Union, the visit of the Bulgarian president in these countries.

During the error analysis, we noticed that in the context of the political parties, SDS appeared many times in with the names of the political leader or the representatives of the BSP party and vice versa. This impeded LSA's classification, because of the similar context.

Among all categories, RIVER and MOUNTAIN obtained the best performances. The rivers Dunav and Maritsa reached 80%, while the mountains Rodopi achieved 81.76% f-score. Looking at the discrimination results for the other names in these categories, it can be seen that their performances are much higher compared to the names of the CITY, COUNTY and PARTY categories. This experiment shows that the discrimination power is related to the type of the NE category we want to resolve.

6 Conclusions

In this paper, we have presented a language independent approach for person name categorization and discrimination. This approach is based on the sentence semantic similarity information derived from LSA. The approach is evaluated with different NE examples for the Spanish and Bulgarian languages. We have observed the discrimination performance of LSA not only with the SINGER and PRESIDENT companies, but also with the CITY, COUNTRY, MOUNTAIN, RIVER and PARTY. This is the first approach which focuses on the resolution of these categories for the Bulgarian language.

The obtained results both for Spanish and Bulgarian are very promising. The baselines are outperformed with 25%. The person fine-grained categorization reaches 90% while the name discrimination varies from 42% to 90%. This variability is related to the degree of the name ambiguity among the conflated names and similar behaviour is observed in the co-occurence approach of (Pedersen et al., 2005).

During the experimental evaluation, we found out that the 100% name purity (e.g. that one name belongs only to one and the same semantic category) which we accept during the data creation in reality contains 9% noise. These observations are confirmed in the additional experimental study we have conducted with the Bulgarian language. According to the obtained results, our text semantic similarity approach performs very well and practically there is no restrain to be adapted to other languages, data sets or even new categories.

7 Future Work

In the future, we want to relate the name discrimination and categorization processes, by first encountering the different underlying meanings of a name and then grouping together the sentences that belong to the same semantic category. This process will increase the performance of the NE fine-grained categorization, and will reduce the errors we encountered during the classification of the singers Enrique Iglesias and Madonna. In addition to this experiment, we want to cluster web pages on the basis of name ambiguity. For instance, we want to process the result for the Google's query George Miller, and form three separate clusters obtained on the basis of a fine-grained and name discrimination. Thus we can form the clusters for George Miller the congressman, the movie director and the father of WordNet. This study will include also techniques for automatic cluster stopping.

Moreover, LSA's ability of language independence can be exploited to resolve cross-language NE categorization and discrimination from which we can extract cross-language pairs of semantically related words characterizing a person e.g. George Bush is seen with White House in English, la Casa Blanca in Spanish, a Casa Branka in Portuguese and Beliat Dom in Bulgarian.

With LSA, we can also observe the time consistency property of a person which changes its semantic category across time. For instance, a student turns into a PhD student, teaching assistant and then university professor, or as in the case of Arnold Schwarzenegger from actor to governor.

Acknowledgements

We would like to thank the three anonymous reviewers for their useful comments and suggestions. This work was partially funded by the European Union under the project QALLME number FP6 IST-033860 and by the Spanish Ministry of Science and Technology under the project TEX-MESS number TIN2006-15265-C06-01.

References

A. Bagga and B. Baldwin. 1998. Entity-based cross-document coreferencing using the vector space model. In *Proceedings of the Thirty-Sixth Annual Meeting of the ACL and Seventeenth International Conference on Computational Linguistics*, pages 79–85.

G. Cleuziou, L. Martin, and C. Vrain. 2004. Poboc: An overlapping clustering algorithm, application to rule-based classification and textual data. In *ECAI*, pages 440–444.

S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman. 1990. Indexing by latent semantic analysis. In *Journal of the American Society for Information Science*, volume 41, pages 391–407.

S. Dumais. 1995. Using lsi for information filtering: Trec-3 experiments. In *The Third Text Retrieval Conference (TREC-3)*, pages 219–230.

M. Fleischman and E. Hovy. 2002. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics*, pages 1–7.

Z. Kozareva, O. Ferrándeza, A. Montoyo, R. Muñoz, A. Suárez, and J. Gómez. 2007. Combining data-driven systems for improving named entity recognition. *Data and Knowledge Engineering*, 61(3):449–466, June.

G. Mann. 2002. Fine-grained proper noun ontologies for question answering. In *COLING-02 on SEMANET*, pages 1–7.

G. Miller and W. Charles. 1991. Contextual correlates of semantic similarity. In *Language and Cognitive Processes*, pages 1–28.

M. Pasca. 2004. Acquisition of categorized named entities for web search. In *CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management*, pages 137–145.

T. Pedersen, A. Purandare, and A. Kulkarni. 2005. Name discrimination by clustering similar contexts. In *CICLing*, pages 226–237.

P. Shah, D. Schneider, C. Matuszek, R.C. Kahlert, B. Aldag, D. Baxter, J. Cabral, M. Witbrock, and J. Curtis. 2006. Automated population of cyc: Extracting information about named-entities from the web. In *Proceedings of the Nineteenth International FLAIRS Conference*, pages 153–158.

H. Shütze. 1998. Automatic word sense discrimination. In *Journal of computational linguistics*, volume 24.

H. Tanev and B. Magnini. 2006. Weakly supervised approaches for ontology population. In *Proceeding of 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 17–24.

Lemmatization of Polish Person Names

Jakub Piskorski
European Commission
Joint Research Centre
Via Fermi 1
21020 Ispra, Italy
Jakub.Piskorski@jrc.it

Marcin Sydow
Polish-Japanese Institute
of Information Technology
Koszykowa 86
02-008 Warsaw, Poland
msyd@pjwstk.edu.pl

Anna Kupść
Université Paris3/LLF, PAS ICS
Case Postale 7031
2, place Jussieu
75251 Paris Cedex 05
akupsc@univ-paris3.fr

Abstract

The paper presents two techniques for lemmatization of Polish person names. First, we apply a rule-based approach which relies on linguistic information and heuristics. Then, we investigate an alternative knowledge-poor method which employs string distance measures. We provide an evaluation of the adopted techniques using a set of newspaper texts.

1 Introduction

Proper names constitute a significant part of natural language texts (estimated to about 10% in newspaper articles) and are important for NLP applications, such as Information Extraction, which rely on automatic text understanding.[1] In particular, coreference resolution (e.g., identifying several name variants as referring to the same entity) plays a crucial role in such systems. Although automatic recognition of proper names in English, French and other major languages has been in the research focus for over a decade now, cf. (Bikel et al., 1997), (Borthwick, 1999), (Li et al., 2003), only a few efforts have been reported for Slavic languages, cf. (Cunningham et al., 2003) (Russian and Bulgarian), (Piskorski, 2005) (Polish). Rich inflection and a more relaxed word order make recognition of proper names in Slavic more difficult than for other languages. Moreover, inflection of proper names is usually quite different from common nouns, which complicates the lemmatization process necessary for correct coreference resolution. In this paper, we focus on lemmatization of Polish person names, the most idiosyncratic class of proper names in this language. First, we report results of a rule-based symbolic approach. We apply different heuristics, mostly based on the internal (morphological and syntactic) structure of proper names but also on the surrounding context. Sometimes, however, the required information is not available, even if the entire document is considered, and lemmatization cannot be performed. Therefore, we experimented with various knowledge-poor methods, namely string distance metrics, in order to test their usefulness for lemmatization of Polish person names as an alternative technique, especially for cases where document-level heuristics are insufficient.

Lemmatization of proper names in Slavic has not attracted much attention so far but some work has been done for Slovene: (Erjavec et al., 2004) present a machine-learning approach to lemmatization of unknown single-token words, whereas (Pouliquen et al., 2005) report on a shallow approach to find base forms.

The organization of the paper is as follows. First, we present a description of phenomena which make lemmatization of Polish person names a difficult task. Next, a rule-based approach and its evaluation are presented. Then, various string distance metrics are introduced, followed by the results of experiments on newspaper texts. The final section presents conclusions and perspectives for future work.

[1] The research presented in this paper was partially founded by the Ministry of Education and Science (Poland), grant number 3T11C00727.

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 27–34,
Prague, June 2007.

case	male name	female name
nom	*Kazimierz Polak*	*Kazimiera Polak*
gen	*Kazimierza Polaka*	*Kazimiery Polak*
dat	*Kazimierzowi Polakowi*	*Kazimierze Polak*
acc	*Kazimierza Polaka*	*Kazimierę Polak*
ins	*Kazimierzem Polakiem*	*Kazimierą Polak*
loc	*Kazimierzu Polaku*	*Kazimierze Polak*
voc	*Kazimierzu Polaku*	*Kazimiero Polak*

Table 1: Declension of Polish male vs. female names

case	sg	pl	sg	pl
nom	*gołąb*	*gołębie*	*Gołąb*	*Gołąbowie*
gen	*gołębia*	*gołębi*	*Gołąba*	*Gołąbów*
dat	*gołębiowi*	*gołębiom*	*Gołąbowi*	*Gołąbom*
acc	*gołębia*	*gołębie*	*Gołąba*	*Gołąbów*
ins	*gołębiem*	*gołębiami*	*Gołąbem*	*Gołąbami*
loc	*gołębia*	*gołębie*	*Gołąbiu*	*Gołąbach*
voc	*gołębiu*	*gołębie*	*Gołąb*	*Gołąbowie*

Table 2: Common noun vs. person name inflection

2 Declension Patterns of Polish Person Names

Polish is a West Slavic language with rich nominal inflection: nouns and adjectives are inflected for case, number and gender. There are 7 cases, 2 numbers and traditionally 3 genders are distinguished: masculine, feminine and neuter. Just like common nouns, Polish person names undergo declension but their inflectional patterns are more complicated. A typical Polish name consists of a first name and a last name; unlike in Russian or Bulgarian, there are no patronymics. Additionally, titles (e.g., *dr* 'Phd', *inż.* 'engineer', *prof.* 'professor') or honorific forms (*pan* 'Mr.' or *pani* 'Mrs./Miss') are often used. In general, both the first and the last name can be inflected, e.g., *Jan Kowalski* (nominative) vs. *Jana Kowalskiego* (genitive/accusative). If the surname is also a regular word form, things get more complicated. Whether the last name can be inflected in such cases depends on several factors, e.g., on the gender of the first name, a category (part-of-speech) and gender of the (common) word used as a surname. For instance, if the surname is a masculine noun, it is inflected only if the first name is also masculine. This is illustrated in Table 1 with declension of the male name *Kazimierz Polak* 'Casimir Pole' and its variant with the female first name *Kazimiera*.

If the surname is an adjective (e.g., *Niski* 'Short'), it is inflected (according to the adjectival paradigm) and agrees in gender with the first name, i.e., male and female last name forms are different (e.g., *Niski* 'Short' (masc.) vs. *Niska* 'Short' (fem.)). The declension of foreign surnames may strongly depend on their origin, and in particular on the pronunciation. For example, the name *Wilde* is pronounced differently in English and German, which impacts its declension in Polish. If it's of English origin, a nominal declension is applied, i.e., *Wilde'a* (gen.), whereas if it comes from German, an adjective-like declension is adopted: *Wildego* (gen.).

Declension of surnames which are also common nouns can be different from the declension of common nouns.[2] In Table 2, we present a comparison of the common noun *gołąb* 'dove' in singular and plural with the corresponding forms used for the surname. A comprehensive overview of this rather intriguing declension paradigm of Polish names is given in (Grzenia, 1998).

Finally, first name forms present problems as well. Foreign masculine first names, whose pronounced version ends in a consonant or whose written version ends in *-a*, *-o*, *-y* or *-i* do in general get inflected (e.g., *Jacques* (nom.) vs. *Jacques'a* (gen./acc.)), whereas names whose pronounced version ends in a vowel and are stressed on the last syllable (e.g., *François*) usually do not change form. For female first names created from a male first name, e.g., *Józef* (masc.) vs. *Józefa* (fem.), there is a frequent homonymy between the nominative form of the female name and the genitive/accusative form of the corresponding male form, e.g., *Józefa* is nominative of *Józefa* (fem.) and genitive/accusative of *Józef* (masc.).

3 Rule-Based Approach to Person Name Lemmatization

3.1 Experiment

Our rule-based approach to person name lemmatization exploits existing resources (a dictionary of first names and contextual triggers) and relies on contextual information (heuristics). It has been implemented using SProUT, a shallow processing platform, integrated with a Polish morphological anal-

[2] The declension of such surnames depends on the local tradition and sometimes can be identical with the pattern used for common nouns.

yser (Piskorski et al., 2004). For first names, all inflected forms of the most frequent Polish first names are stored in a database so a simple gazetteer look-up associates names with the corresponding base form. We also used a list of ca 30 000 foreign first names (nominative forms). For last names, we applied several heuristic rules in order to recognize and produce their base forms. First, we identify most common types of Polish surnames, e.g., capitalized words ending in *-skiego*, *-skim*, *-skiemu* or *-icza*, *-iczem*, *-iczu* (typical last name suffixes), and convert them to the corresponding base forms (i.e., words ending in *-ski* and *-icz*, respectively). In this way, a significant number of names can be lemmatized in a brute-force manner.

For all remaining surnames, more sophisticated rules have to be applied. As discussed in sec. 2, these rules have to take into account several pieces of information such as part-of-speech and gender of the (common) word which serves as a surname, but also gender of the first name. The major problem we encountered while applying these rules is that the information necessary to trigger the appropriate rule is often missing. For example, in sentence (1), inferring gender of the surname/first name could involve a subcategorization frame for the verb *powiadomić* 'inform', which requires an accusative NP argument. In this way we might possibly predict that the base form of *Putina* is *Putin*, as *-a* is the typical accusative ending of masculine names. Since the subcategorization lexicon is not available, such instances are either not covered or different heuristics are employed for guessing the base form.

(1) *Powiadomiono wczoraj wieczorem V. Putina o ataku.*
informed yesterday evening V. Putin$_{acc}$ about attack
'Yesterday evening they informed V. Putin about the attack.'

Additionally, grammar rules may produce variants of recognized full person names. For example, for the full name *CEO dr Jan Kowalski* the following variants can be produced: *Kowalski*, *CEO Kowalski*, *dr Kowalski*, etc. As the grammar rules always return the longest match, a shorter form may not be recognized. The produced variants are therefore used in the second pass through the text in order to identify 'incomplete' forms. As no morphological generation is involved, only base forms can be identified in this way. The system evaluation indicates that 23.8% of the recognized names were identified by this partial coreference resolution mechanism.

An analysis of incorrectly recognized named entities (NEs) revealed that major problems concerned (a) classical ambiguities, such as a proper name vs. a common word, and (b) person vs. organization name, caused by a specific word order and a structural ambiguity of phrases containing NEs. Let us consider the following examples to illustrate the problems.

(2) *Dane Federalnego Urzędu Statystycznego*
Data$_{nom}$ federal$_{gen}$ office$_{gen}$ statistical$_{gen}$
'Data of the federal office for statistics'

(3) *prezes Della*
president$_{nom}$ Dell$_{gen}$
'president of Dell'

(4) *kanclerz Austriaków*
chancellor$_{nom}$ Austrians$_{gen}$
'chancellor of the Austrians'

(5) *... powiedział prezes spółki Kruk*
said president$_{nom}$ company$_{gen}$ Kruk$_{nom}$
'. . . said the president of Kruk company / Kruk, the president of the company'

The text fragment *Dane Federalnego* in (2) is recognized by the grammar as a person name since *Dane* is a gazetteer entry for a foreign (English) first name. Consequently, *Federalnego Urzędu Statystycznego* could not be recognized as an organization name. Potentially, heuristics solving such NE overlapping collisions could improve the precision. Similar techniques have been applied to other languages. In (3) and (4) the names *Della* 'of Dell' and *Austriaków* 'of Austrians' were erroneously recognized as surnames. The rule matching a token representing a title followed by a capitalized word, adopted for English person names, is less reliable for Polish due to declension of proper names and lack of prepositions in genitive constructions. One solution to this problem would involve matching *Della* and *Austriaków* with their base forms (*Dell* and *Austriacy*, resp.), which might appear in the immediate context. In this way, the name type could be validated. However, a corpus inspection revealed that quite frequently no base form appears in the same document. The last example, (5), illustrates another problem, which is even harder to solve. The phrase *prezes* *spółki Kruk* is structurally ambiguous, i.e., it can be bracketed as [*prezes* [*spółki Kruk*]] or [[*prezes spółki*] *Kruk*]. Consequently, the name *Kruk* might either refer to a company name ('...said the president of the Kruk company') or to a person name ('...said Kruk, the president of the company'). Inferring the proper interpretation might not be possible even if we consider the subcategorization frame of the verb *powiedzieć* 'to say'.

3.2 Evaluation

For evaluation of recognition and lemmatization of person names, a set of 30 articles on various topics (politics, finance, sports, culture and science) has been randomly chosen from *Rzeczpospolita* (Weiss, 2007), a leading Polish newspaper. The total number of person name occurrences in this document set amounts to 858. Evaluation of recognition's precision and recall yielded 88.6% and 82.6%, respectively. Precision of lemmatization of first names and surnames achieved 92.2% and 75.6%, respectively. For 12.4% of the recognized person names more than one output structure was returned. For instance, in case of the person name *Marka Belki*, the first name *Marka* is interpreted by the gazetteer either as an accusative form of the male name *Marek* or as a nominative form of a foreign female name *Marka*. In fact, 10% of the Polish first-name forms in our gazetteer are ambiguous with respect to gender. As for the last name *Belki*, it is a genitive form of the common Polish noun *belka* 'beam', so the base form can be obtained directly. Nevertheless, as inflection of proper names differs from that of common nouns, various combinations of the regular noun *Belka* and the special proper name form *Belki* are possible, which increases ambiguity of the identified form. All possible lemmatizations are as follows:

(6) *Marek Belka* (masc.),
Marka Belka (fem.),
Marek Belki (masc.),
Marka Belki (fem.)

A good heuristics to reduce such ambiguous lemmatizations is to prioritize rules which refer to morphological information over those which rely solely on orthography and/or token types.

4 Application of String Distance Metrics for Lemmatization

Since knowledge-based lemmatization of Polish NEs is extremely hard, we also explored a possibility of using string distance metrics for matching inflected person names with their base forms (and their variants) in a collection of document, rather than within a single document. The rest of this section describes our experiments in using different string distance metrics for this task, inspired by the work presented in (Cohen et al., 2003) and (Christen, 2006).

The problem can be formally defined as follows. Let A, B and C be three sets of strings over some alphabet Σ, with $B \subseteq C$. Further, let $f : A \rightarrow B$ be a function representing a mapping of inflected forms (A) into their corresponding base forms (B). Given, A and C (the search space), the task is to construct an approximation of f, namely $\widehat{f} : A \rightarrow C$. If $\widehat{f}(a) = f(a)$ for $a \in A$, we say that $\widehat{f}$ returns the correct answer for a; otherwise, $\widehat{f}$ is said to return an incorrect answer. For another task, a multi-result experiment, we construct an approximation $f^* : A \rightarrow 2^C$, where f^* returns the correct answer for a if $f(a) \in f^*(a)$.

4.1 String distance metrics

In our experiments, we have explored mainly character-level string metrics[3] applied by the database community for record linkage.

Our point of departure is the well-known *Levenshtein* edit distance metric specified as the minimum number of character-level operations (insertion, deletion or substitution) required for transforming one string into another (Levenshtein, 1965) and *bag distance* metric (Bartolini et al., 2002) which is a time-efficient approximation of the *Levenshtein* metric. Next, we have tested the *Smith-Waterman* (Smith and Waterman, 1981) metric, which is an extension of *Levenshtein* metric and allow a variable cost adjustment to edit operations and an alphabet mapping to costs.

Another group of string metrics we explored is based on a comparison of character-level *n*-grams in two strings. The *q-gram* metric (Ukkonen, 1992) is

[3] Distance (similarity) metrics map a pair of strings s and t to a real number r, where a smaller (larger) value of r indicates greater (lower) similarity.

computed by counting the number of q-grams contained in both strings. An extension to q-grams is to add positional information, and to match only common q-grams that occur at a specified distance from each other (*positional q-grams*) (Gravano et al., 2001). Finally, the *skip-gram* metric (Keskustalo et al., 2003) is based on the idea that in addition to forming bigrams of adjacent characters, bigrams that skip characters are considered as well. *Gram classes* are defined that specify what kind of skip-grams are created, e.g. $\{0, 1\}$ class means that regular bigrams (0 characters skipped) and bigrams that skip one character are formed. We have explored $\{0, 1\}$, $\{0, 2\}$ and $\{0, 1, 2\}$ gram classes.

Taking into account the Polish declension paradigm, we also added a basic metric based on the longest common prefix, calculated as follows:

$$CP_\delta(s,t) = ((|lcp(s,t)| + \delta)^2/(|s| \cdot |t|),$$

where $lcp(s,t)$ denotes the longest common prefix for s and t. The symbol δ is a parameter for favoring certain suffix pairs in s (t). We have experimented with two variants: CP_{δ_1} with $\delta = 0$ and CP_{δ_2}, where δ is set to 1 if s ends in: *o*, *y*, *ą*, *ę*, and t ends in an *a*, or 0 otherwise. The latter setting results from empirical study of the data and the declension paradigm.

For coping with multi-token strings, we tested a similar metric called *longest common substrings* (LCS) (Christen, 2006), which recursively finds and removes the longest common substring in the two strings compared, up to a specified minimum length. Its value is calculated as the ratio of the sum of all found longest common substrings to the length of the longer string. We extended LCS by additional weighting the lengths of the longest common substrings. The main idea is to penalize the longest common substrings which do not match the beginning of a token in at least one of the compared strings. In such cases, the weight for $lcs(s,t)$ (the longest common substring for s and t) is computed as follows. Let α denote the maximum number of non-whitespace characters which precede the first occurrence of $lcs(s,t)$ in s or t. Then, $lcs(s,t)$ is assigned the weight:

$$w_{lcs(s,t)} = \frac{|lcs(s,t)| + \alpha - \max(\alpha, p)}{|lcs(s,t)| + \alpha}$$

where p has been experimentally set to 4. We refer to the 'weighted' variant of LCS as $WLCS$.

Good results for name-matching tasks (Cohen et al., 2003) have been reported using the *Jaro* metric and its variant, the *Jaro-Winkler* (JW) metric (Winkler, 1999). These metrics are based on the number and order of common characters in two compared strings. We have extended the *Jaro-Winkler* metric to improve the comparison of multi-token strings. We call this modification JWM and it can be briefly characterized as follows. Let $J(s,t)$ denote the value of the *Jaro* metric for s and t. Then, let $s = s_1 \ldots s_K$ and $t = t_1 \ldots t_L$, where s_i (t_i) represent i-th token of s and t respectively, and assume, without loss of generality, $L \leq K$. $JWM(s,t)$ is defined as:

$$JWM(s,t) = J(s,t) + \delta \cdot boost_p(s,t) \cdot (1 - J(s,t))$$

where δ denotes the common prefix adjustment factor and $boost_p$ is calculated as follows:

$$boost_p(s,t) = \frac{1}{L} \cdot \sum_{i=1}^{L-1} \min(|lcp(s_i,t_i)|, p) + \frac{\min(|lcp(s_L, t_L..t_K)|, p)}{L}$$

The main idea behind JWM is to boost the *Jaro* similarity for strings with the highest number of agreeing initial characters in the corresponding tokens in the compared strings.

Finally, for multi-token strings, we tested a recursive matching pattern, known also as *Monge-Elkan* distance (Monge and Elkan, 1996). The intuition behind this measure is the assumption that a token in s (strings are treated as sequences of tokens) corresponds to a token in t which has the highest number of agreeing characters. The similarity between s and t is the mean of these maximum scores. Two further metrics for multi-token strings were investigated, namely *Sorted-Tokens* and *Permuted-Tokens*. The first one is computed in two steps: (a) first, tokens forming a full string are sorted alphabetically, and then (b) an arbitrary metric is applied to compute the similarity for the 'sorted' strings. The latter compares all possible permutations of tokens forming the full strings and returns the calculated maximal similarity value.

A detailed description of string metrics used here is given in (Christen, 2006) and in (Piskorski et al., 2007).

4.2 Test Data

For the experiments on coreference of person names, we used two resources: (a) a lexicon of the most frequent Polish first names (PL-F(IRST)-NAMES) consisting of pairs of an inflected form and the corresponding base form, and (b) an analogous lexicon of inflected full person names (first name + surname) (PL-FULL-NAMES).[4] The latter resource was created semi-automatically as follows. We have automatically extracted a list of 22485 full person-name candidates from a corpus of 15724 on-line news articles from *Rzeczpospolita* by using PL-F-NAMES lexicon and an additional list of 30000 uninflected foreign first names. Subsequently, we have randomly selected a subset of about 1900 entries (inflected forms) from this list.

In basic experiments, we simply used the base forms as the search space. Moreover, we produced variants of PL-F-NAMES and PL-FULL-NAMES by adding to the search space base forms of foreign first names and a complete list of full names extracted from the *Rzeczpospolita* corpus, respectively. Table 3 gives an overview of our test datasets.

Dataset	**#inflected**	**#base**	**search space**
PL-F-NAMES	5941	1457	1457
PL-F-NAMES-2	5941	1457	25490
PL-FULL-NAMES	1900	1219	1219
PL-FULL-NAMES-2	1900	1219	2351
PL-FULL-NAMES-3	1900	1219	20000

Table 3: Dataset used for the experiments

4.3 Evaluation Metrics

Since for a given string more than one answer can be returned, we measured the accuracy in three ways. First, we calculated the accuracy on the assumption that a multi-result answer is incorrect and we defined *all-answer accuracy* (AA) measure which penalizes multi-result answers. Second, we measured the accuracy of single-result answers (*single-result accuracy* (SR)) disregarding the multi-result answers. Finally, we used a weaker measure which treats a multi-result answer as correct if one of the results in the answer set is correct (*relaxed-all-answer accuracy* (RAA)).

Let s denote the number of strings for which a single result (base form) was returned. Analogously, m is the number of strings for which more than one result was returned. Let s_c and m_c denote, respectively, the number of correct single-result answers returned and the number of multi-result answers containing at least one correct result. The accuracy metrics are computed as: $AA = s_c/(s+m)$, $SR = s_c/s$, and $RAA = (s_c + m_c)/(s+m)$.

4.4 Experiments

We started our experiments with the PL-F-NAME dataset and applied all but the multi-token strings distance metrics. The results of the accuracy evaluation are given in Table 4. The first three columns give the accuracy figures, whereas the column labeled **AV** gives an average number of results returned in the answer set.

Metrics	**AA**	**SR**	**RAA**	**AV**
Bag Distance	0.476	0.841	0.876	3.02
Levenshtein	0.708	0.971	0.976	2.08
Smith-Waterman	0.625	0.763	0.786	3.47
Jaro	0.775	0.820	0.826	2.06
Jaro-Winkler	0.820	0.831	0.831	2.03
q-grams	0.714	0.974	0.981	2.09
pos q-grams	0.721	0.976	0.982	2.09
skip grams	0.873	0.935	0.936	2.14
LCS	0.696	0.971	0.977	12.69
WLCS	0.731	**0.983**	**0.986**	2.97
CP_{δ_1}	0.829	0.843	0.844	2.11
CP_{δ_2}	**0.947**	0.956	0.955	2.18

Table 4: Results for PL-F-NAMES

Interestingly, the simple linguistically-aware common prefix-based measure turned out to work best in the **AA** category, which is the most relevant one, whereas *WLCS* metrics is the most accurate in case of single-result answers and the **RAA** category. Thus, a combination of the two seems to be a reasonable solution to further improve the performance (i.e., if *WLCS* provides a single answer, return this answer, otherwise return the answer of CP_{δ_2}). Next, the time-efficient *skip grams* metrics performed surprisingly well in the **AA** category. This result was achieved with $\{0, 2\}$ gram classes. Recall that about 10% of the inflected first name forms in Polish are ambiguous, as they are either a male or a female person name, see sec. 2.

Clearly, the **AA** accuracy figures in the experiment run on the PL-F-NAME-2 (with a large search space) was significantly worse. However, the **SR**

[4] Inflected forms which are identical to their corresponding base form were excluded from the experiments since finding an answer for such cases is straightforward.

accuracy for some of the metrics is still acceptable. The top ranking metrics with respect to **SR** and **AA** accuracy are given in Table 5. Metrics which return more than 5 answers on average were excluded from this list. Also in the case of PL-F-NAME-2 the combination of $WLCS$ and CP_{δ_2} seems to be the best choice.

Metrics	SR	AA
WLCS	**0.893**	0.469
CP_{δ_2}	0.879	**0.855**
pos 2-grams	0.876	0.426
skip grams	0.822	0.567
2-grams	0.810	0.398
LCS	0.768	0.340
CP_{δ_1}	0.668	0.600
JW	0.620	0.560

Table 5: Top results for PL-F-NAMES-2

Finally, we have made experiments for full person names, each represented as two tokens. It is important to note that the order of the first name and the surname in some of the entities in our test datasets is swapped. This inaccuracy is introduced by full names where the surname may also function as a first name. Nevertheless, the results of the experiment on PL-FULL-NAMES given in Table 6 are nearly optimal. JWM, $WLCS$, LCS, skip grams and *Smith-Waterman* were among the 'best' metrics.

Internal Metrics	AA	SR	RAA	AV
Bag Distance	0.891	0.966	0.966	3.13
Smith-Waterman	0,965	0,980	0,975	3,5
Levenshtein	0.951	0.978	0.970	4.59
Jaro	0.957	0.970	0.964	3.54
JW	0.952	0.964	0.958	3.74
JWM	0.962	0.974	0.968	3.74
2-grams	0.957	0.988	0.987	3.915
pos 3-grams	0.941	0.974	0.966	4.32
skip-grams	0.973	0.991	0.990	5.14
LCS	0.971	0.992	0.990	5.7
WLCS	**0.975**	**0.993**	**0.992**	6.29

Table 6: Results for PL-FULL-NAMES

The *Monge-Elkan*, *Sorted-Tokens* and *Permuted-Tokens* scored in general only slightly better than the basic metrics. The best results oscillating around 0.97, 0.99, and 0.99 for the three accuracy metrics were obtained using *LCS*, *WLCS*, *JWM* and CP_δ metrics as internal metrics. The highest score was achieved by applying *Sorted-Tokens* with *JWM* with 0.976 in **AA** accuracy.

Further, in order to get a better picture, we have compared the performance of the aforementioned 'recursive' metrics on PL-FULL-NAMES-2, which has a larger search space. The most significant results for the **AA** accuracy are given in Table 7. The JWM metric seems to be the best choice as an internal metric, whereas $WLCS$, CP_{δ_2} and *Jaro* perform slightly worse.

Internal M.	Monge-Elkan	Sorted-Tokens	Permuted-Tokens
Bag Distance	0.868	0.745	0.745
Jaro	0.974	0.961	0.968
JWM	**0.976**	**0.976**	**0.975**
SmithWaterman	0.902	0.972	0.967
3-grams	0.848	0.930	0.911
pos 3-grams	0.855	0.928	0.913
skip-grams	0.951	0.967	0.961
LCS	0.941	0.960	0.951
WLCS	0.962	0.967	0.967
CP_{δ_1}	0.969	n.a.	n.a.
CP_{δ_2}	0.974	n.a.	n.a.

Table 7: **AA** accuracy for PL-FULL-NAMES-2

In our last experiment we selected the 'best' metrics so far and tested them against PL-FULL-NAMES-3 (largest search space). The top results for non-recursive metrics are given in Table 8. JWM and $WLCS$ turned out to achieve the best scores.

Metrics	AA	SR	RAA	AV
Levenshtein	0.791	0.896	0.897	2.20
Smith-Waterman	0.869	0.892	0.889	2.35
JW	0.791	0.807	0.802	2.11
JWM	**0.892**	0.900	0.901	2.11
skip-grams	0.852	0.906	0.912	2.04
LCS	0.827	0.925	0.930	2.48
WLCS	0.876	**0.955**	**0.958**	2.47

Table 8: Results for PL-FULL-NAMES-3

The top scores achieved for the recursive metrics on PL-FULL-NAMES-3 were somewhat better. In particular, *Monge-Elkan* performed best with CP_{δ_2} (0.937 **AA** and 0.946 **SR**) and slightly worse results were obtained with *JWM*. *Sorted-Tokens* scored best in **AA** and **SR** accuracy with *JWM* (0.904) and *WLCS* (0.949), respectively. Finally, for *Permuted-Tokens* the identical setting yielded the best results, namely 0.912 and 0.948, respectively.

5 Conclusions and Perspectives

For Slavic languages, rich and idiosyncratic inflection of proper names presents a serious problem for lemmatization. In this paper we investigated two different techniques for finding base forms of person names in Polish. The first one employs heuristics and linguistic knowledge. This method does not provide optimal results at the moment as necessary tools and linguistic resources, e.g., a morphological generator or a subcategorization lexicon, are still underdeveloped for Polish. Moreover, contextual heuristics do not always find a solution as the required information might not be present in a single document. Therefore, we considered string distance metrics as an alternative approach. The results of applying various measures indicate that for first names, simple common prefix (CP$_\delta$) metric obtains the best results for all-answer accuracy, whereas the weighted longest common substrings ($WLCS$) measure provides the best score for the single-result accuracy. Hence, a combination of these two metrics seems the most appropriate knowledge-poor technique for lemmatizing Polish first names. As for full names, our two modifications ($WLCS$ and JWM) of standard distance metrics and CP$_\delta$ obtain good results as internal metrics for recursive measures and as stand-alone measures.

Although the results are encouraging, the presented work should not be considered a final solution. We plan to experiment with the best scoring metrics (e.g., for **AA** and **SR**) in order to find optimal figures. Additionally, we consider combining the two techniques. For example, string distance metrics can be used for validation of names found in the context. We also envisage applying the same methods to other types of proper names as well as to lemmatization of specialized terminology.

References

I. Bartolini, P. Ciacca, and M. Patella. 2002. String matching with metric trees using an approximate distance. In *Proceedings of SPIRE*, LNCS 2476, Lisbon, Portugal.

D. Bikel, S. Miller, R. Schwartz, and R. Weischedel. 1997. Nymble: A High-performance Learning Name-finder. In *Proceedings of ANLP-1997*, Washington DC, USA.

A. Borthwick. 1999. A Maximum Entropy Approach to Named Entity Recognition. PhD Thesis, Department of Computer Science, New York University.

P. Christen. 2006. A Comparison of Personal Name Matching: Techniques and Practical Issues. Technical report, TR-CS-06-02, Computer Science Laboratory, The Australian National University, Canberra, Australia.

W. Cohen, P. Ravikumar, and S. Fienberg. 2003. A comparison of string metrics for matching names and records. In *Proceedings of the KDD2003*.

H. Cunningham, E. Paskaleva, K. Bontcheva, and G. Angelova. 2003. Information extraction for Slavonic languages. In *Proceedings of the Workshop IESL*, Borovets, Bulgaria.

T. Erjavec and S. Džeroski. 2004. Machine Learning of Morphosyntactic Structure: Lemmatising Unknown Slovene Words. In *Journal of Applied Artificial Intelligence*, 18(1), pages 17-40.

L. Gravano, P. Ipeirotis, H. Jagadish, S. Koudas, N. Muthukrishnan, L. Pietarinen, and D. Srivastava. 2001. Using q-grams in a DBMS for Approximate String Processing. *IEEE Data Engineering Bulletin*, 24(4):28–34.

J. Grzenia. 1998. *Słownik nazw własnych — ortografia, wymowa, słowotwórstwo i odmiana*. PWN.

H. Keskustalo, A. Pirkola, K. Visala, E. Leppanen, and K. Jarvelin. 2003. Non-adjacent digrams improve matching of cross-lingual spelling variants. In *Proceedings of SPIRE*, LNCS 22857, Manaus, Brazil, pages 252–265.

V. Levenshtein. 1965. Binary Codes for Correcting Deletions, Insertions, and Reversals. *Doklady Akademii Nauk SSSR*, 163(4):845–848.

W. Li, R. Yangarber, and R. Grishman. 2003. Bootstrapping Learning of Semantic Classes from Positive and Negative Examples. In *Proceedings of the ICML-2003 Workshop on The Continuum from Labeled to Unlabeled Data*.

A. Monge and C. Elkan. 1996. The Field Matching Problem: Algorithms and Applications. In *Proceedings of Knowledge Discovery and Data Mining 1996*, pages 267–270.

J. Piskorski, P. Homola, M. Marciniak, A. Mykowiecka, A. Przepiórkowski, and M. Woliński. 2004. Information Extraction for Polish Using the Sprout Platform. *Proceedings of ISMIS 2004, Zakopane*.

J. Piskorski. 2005. Named-entity Recognition for Polish with SProUT. In *Proceedings of IMTCI 2004*, LNCS Vol 3490, Warsaw, Poland.

J. Piskorski and M. Sydow. 2007. Usability of String Distance Metrics for Name Matching Tasks in Polish. In progress.

B. Pouliquen, R. Steinberger, C. Ignat, I. Temnikova, A. Widiger, W. Zaghouani and J. Žižka. 2005. Multilingual person name recognition and transliteration. *CORELA - Cognition, Représentation, Langage. Numéros spéciaux, Le traitement lexicographique des noms propres*, ISSN 1638-5748.

T. Smith and M. Waterman. 1981. Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197.

E. Ukkonen. 1992. Approximate String Matching with q-grams and Maximal Matches. *Theoretical Computer Science*, 92(1):191–211.

D. Weiss. 2007. Korpus Rzeczpospolitej. Web document: *http://www.cs.put.poznan.pl/dweiss/rzeczpospolita*

W. Winkler. 1999. The state of record linkage and current research problems. Technical report, U.S. Bureau of the Census, Washington, DC.

Automatic Processing of Diabetic Patients' Hospital Documentation

Małgorzata Marciniak
Institute of Computer Science, PAS
Ordona 21, 01-237 Warszawa, Poland
mm@ipipan.waw.pl

Agnieszka Mykowiecka
Institute of Computer Science, PAS
Ordona 21, 01-237 Warszawa, Poland
agn@ipipan.waw.pl

Abstract

The paper presents a rule-based information extraction (IE) system for Polish medical texts. We select the most important information from diabetic patients' records. Most data being processed are free-form texts, only a part is in table form. The work has three goals: to test classical IE methods on texts in Polish, to create relational database containing the extracted data, and to prepare annotated data for further IE experiments.

1 Introduction

Information extraction from natural language text has become an important task for NLP applications during the last years. In the era of huge text data collections, these methods allows us to perform searches within reasonable time. For the purpose of IE, many methods based on very different approaches (formal grammars, statistics, artificial intelligence) have already been elaborated. In spite of a great number of described experiments, the invented methods were untested on Polish texts. Nowadays, there is great interest in statistical and machine learning methods, e.g. (Bunescu et al., 2003), but applying machine learning techniques to Polish texts is difficult, as there are hardly any annotated Polish data (excluding morpho-syntactic information which is available). The second obstacle was the type of chosen data – relatively low number of available records with complex text. That is why we decided to carry out a rule-based IE system.[1] Below, we present the system selecting information from diabetic patients' hospital records written in unrestricted Polish. We defined a domain dependent set of rules basing on an expert's knowledge and tested them on the previously unseen reports. The extracted data were put into a database allowing for statistical analysis. The other result of the project, the annotated set of original reports, can be further used for applying other methods of IE to these texts.

In our project, we use the SProUT (Shallow Processing with Unification and Typed Feature Structures) system, (Drożdżyński et al., 2004). SProUT is a general purpose platform consisting of a set of components for basic linguistic operations. Grammar rules are regular expressions on typed feature structures (TFS) which are results of tokenization or morphological analysis, as well as information from the domain lexicon. SProUT differs from many other systems in that it allows for unification of TFSs thus allows more general extraction rules. Analysing Polish text is possible due to the integration (Piskorski et al., 2004) of Morfeusz, a morphological analyser for Polish (Woliński, 2006).

Although most biomedical IE activities are related to literature mining and terminology extraction, (e.g. (Bunescu et al., 2003), (Tveit and Saetre 2005)), clinical patients record mining is not a new research goal for all languages, e.g. (Hahn, Romacker and Schulz, 2002). In (Hripcsak et al., 2002) 24 clinical conditions were extracted from narrative chests radiographic reports. The task closest to the pre-

[1] Our first rule-based IE experiment concerned mammography reports (Mykowiecka, Kupść and Marciniak, 2004).

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 35–42,
Prague, June 2007. ©2007 Association for Computational Linguistics

sented here, i.e. searching for information contained in natural language patients' discharge summaries was undertaken in project MENELAS (Zweigenbaum, 1994) and AMBIT (Harkema et al., 2004). In the last one, the extraction rules were based on both syntactic (word category) and semantic information (e.g. latitude-noun or area-noun). 83 radiology reports were processed and descriptions of lung cancers extracted and evaluated. The exemplary results for location were: 61% precision and 92% recall. Results of our experiment are shown in sections 4 and 6.

2 Domain description

For the purpose of diabetic patients' hospital documentation analysis, we elaborated a domain model for the part of data which we are interested in. The model has been defined on the basis of an expert's knowledge and the data i.e. hospital documents. The model describes information on a patient, hospitalisation, diagnosis, tests, treatment and complications. To formalize it, we used OWL-DL standard and the Protégé ontology editor. A part of the diabetes ontology is shown in Fig. 1.

For the purpose of information extraction in SProUT, the ontology had to be translated (manually) into a typed feature structures hierarchy. In the extraction system, the domain model is represented by typed TFSs. A feature's (attribute's) value can be an atomic type, another TFS, or a list of atomic types or TFSs. The type hierarchy contains 139 types with 65 attributes, but as much as 65 types represent medicine terms.

An example of a structure defined to represent basic information about a patient's diabetes is given in Fig. 2. The structure is of the type *diabet_desc_str* and has five attributes. A value of the D_TYPE attribute has the type of *d_type_t* which is a supertype for three types of diabetes: *first, second, other*. The next attribute HBA1C refers to the results of an important diabetes monitoring test. Its numerical value is represented as a string. Next two attributes are of boolean type and indicate if the illness is uncontrolled and if the patient had incidences of hypoglycaemia. A value of the last attribute DIAB_FROM is another TFS of type *diab_from_str* representing when the diabetes have been diagnosed. This information can be given in different ways: in words e.g., *wieloletna* 'long-lasting'; as a date — *w 1990 roku* 'in the year 1990'; relatively *20 lat temu* '20 years ago'; or *w 20 roku życia* 'in the 20th year of life'. All these types of information demand different representation.

- BiochemicalData
    - BloodData
        - HB1C
- Diet
    - DiabetDiet
- DiseaseOrSymptom
    - Disease
        - Alcoholism
        - AutoimmuneDisease
        - Diabetes
            - DiabetesType1
            - DiabetesType2
            - DiabetesTypeOther
    - Symptom
        - Angiopathy
            - Macroangiopathy
            - Microangiopathy
        - BoodSymptom
            - Hypoglicemia
        - DiabeticFood
        - Neuropathy
            - AutonomicNeuropathy
            - PeripheralPolineuropathy
        - UrineSymptom
            - Acetonuria
            - Microalbuminuria
- Hospitalization
- Medicine
    - DiabeticMedicine
        - Insulin
        - OralDiabeticMedicine
- Treatement
    - TreatementScheme

Figure 1: Fragment of the ontology

$$\begin{bmatrix} \textit{diabet\_desc\_str} & \\ \text{D\_TYPE} & \textit{d\_type\_t} \\ \text{HBA1C} & \textit{string} \\ \text{UNCONTROLLED} & \textit{bool\_t} \\ \text{HYPOGLYCAEMIA} & \textit{bool\_t} \\ \text{DIAB\_FROM} & \textit{diab\_from\_str} \end{bmatrix}$$

Figure 2: Structure of type *diabet_desc_str*

Every one document we process concerns one patient's visit in hospital. A particular visit is identified (see Fig. 3) by two parameters: ID number within a year and a year (attribute ID_YEAR). Sometimes some results of tests are available after the patient leaves the hospital. In such cases, there are additional hospital documents referring to these visits described by an attribute CONT: *yes* — continuation.

$$\left[\begin{array}{l} \mathit{id\_str} \\ \text{ID}\ \ \mathit{string} \\ \text{ID\_YEAR}\ \mathit{string} \\ \text{CONT}\ \ \mathit{bool\_t} \end{array}\right]$$

Figure 3: Visit's identification structure

The specific structures are defined for representing the following information:

- identification of a patient's visit in hospital, dates when the hospitalisation took place, and its reasons,
- patient information: identification, age, sex, weight,
- data about diabetes (see Fig. 2),
- complications,
- other illnesses including autoimmunology and accompanying illnesses, which may be correlated with diabetes, like hypertension,
- diabetes treatment: recognition of insulin type and its doses and other oral medications,
- diet: how many calories, and how many meals are recommended,
- patient's education, observing of diet, modification of treatment or diet.

In order to represent complications we defined the appropriate hierarchy. It contains three main types of complications: angiopathy, neuropathy and diabetic foot. The first two have subtypes. Angiopathy divides into micro and macroangiopathy, and neuropathy can be autonomic neuropathy or peripheral polineuropathy. Micro and macroangiopaty has further subtypes. One common complication — rethinopathy is a subtype of microangiopathy and has additional attribute, that represents information about cooccurring maculopathy. Rethinopathy has also subtypes.

Sometimes it is convenient to recognise more then one complication through one rule. In this case, results are represented in a list. For example, the result of recognition of the following phrase describing complications *z neuropatią autonomiczną i obwodową* 'with autonomic and peripheral neuropathy' is represented in Fig. 4. These two complications cannot be identified separately, as there is only one occurrence of the keyword *neuropathy*.

$$\left[\begin{array}{ll} \multicolumn{2}{l}{\mathit{complication\_list}} \\ \text{FIRST} & \mathit{autonomic\_neuropathy} \\ \text{REST} & \left[\begin{array}{l} \mathit{complication\_list} \\ \text{FIRST}\ \mathit{peripheral\_polyneuropathy} \\ \text{REST}\ \mathit{null} \end{array}\right] \end{array}\right]$$

Figure 4: List of complications

3 Information Extraction

3.1 Domain dictionary — gazetteer

A domain dictionary contains all forms of the terms important to the domain terminology. These terms came from the data set or were introduced into the lexicon on the basis of a domain expert's knowledge. The lexicon contains among others all insulin and oral medication names important in diabetology, we introduced forms in nominative and genitive (if such exist) — only these forms appeared in the documents. The other group of words in the dictionary consists of names of diseases and diabetic complications. They have been introduced into the lexicon in all forms used in the documents.

In this specific domain lexicon, there are no information about grammatical categories because it is not used within the grammar rules. In the dictionary, we have only semantic information about entries. There are two levels of semantic information: GTYPE — groups entries with a similar meaning, and G_CONCEPT connects an entry with its unique interpretation. The lexicon is rather small — just over 200 word forms. In Fig. 5, there is a fragment of the gazetteer with eight entries. All of them refer to different types of neuropathy complications.

3.2 Grammar rules

A grammar in SProUT consists of rules, which are regular expressions over TFSs with functional operators and coreferences, representing the recognition

```
neuropatia | GTYPE: gaz_comp | G_CONCEPT: neuropathy_t
Neuropatia | GTYPE: gaz_comp | G_CONCEPT: neuropathy_t
Neuropatią | GTYPE: gaz_comp | G_CONCEPT: neuropathy_t
neuropatią | GTYPE: gaz_comp | G_CONCEPT: neuropathy_t
obwodową | GTYPE: gaz_neuro |
             G_CONCEPT: peripheral_polineuropathy
obwodowa | GTYPE: gaz_neuro |
             G_CONCEPT: peripheral_polineuropathy
autonomiczną | GTYPE: gaz_neuro |
               G_CONCEPT: autonomic_neuropathy
autonomiczna | GTYPE: gaz_neuro |
               G_CONCEPT: autonomic_neuropathy
```

Figure 5: A fragment of gazetteer

pattern. Output structures are also TFSs. Rules use three sources of information: tokenization (structures of type *token* for recognising, among others, abbreviations, dates, numbers), morphological analysis (structures of type *morph*), and a domain dictionary (*gazetteer*).

The SProUT grammar described in the paper consists of about 100 extraction rules. Each rule detects words or phrases describing information presented in section 2. For example, the rule in Fig. 6 recognises the identification number of a patient's visit in hospital. The first line recognises a word from morphological lexicon that has the base form (STEM) *numer* 'number' or an abbreviation[2] of this word, so they are recognised as a token with an appropriate SURFACE form. The next line: `token?` omits a dot after the abbreviation, if it is necessary. Next two lines recognise the keywords with the base forms *księga* ('book', 'document') and *główny* ('main'). Then, the identification number of the document is recognised by the `liczba_nat` rule called (via the `@seek` operator). The number is unified with the value of the ID attribute in the output structure. Next two lines are optional, they recognise a year number after a slash or a backslash, if this information is present. If not, the year is fixed, during postprocessing, according to the dates of the patient's visit in hospital. In this particular case, the value of the attribute CONT is *no* in the output structure. The rule in Fig. 6 captures, among other, the following phrases:

- *Numer księgi głównej 11125/2006*
  'Number of the main document 11125/2006'

```
nr_ksiegi :>
    (morph & [STEM "numer"] | token & [SURFACE "nr"]
                            | token & [SURFACE "Nr"])
    token ?
    morph & [STEM "księga"]
    morph & [STEM "główny"]
    @seek(liczba_nat) & [LICZ #nr]
    ((token & [TYPE slash] | token & [TYPE back_slash])
      @seek (liczba_nat) & [LICZ #nr1])?
->id_str & [ID #nr, ID_YEAR #nr1, CONT no].
```

Figure 6: Visit's identification rule

```
chor_autoimm:>
  (morph & [STEM "choroba"] | morph & [STEM "zespół"])
  gazetteer & [GTYPE gaz_autoimm, G_CONCEPT #type]
->autoimm_dis_str & [AUTOIMM_DISEASE #type].
```

Figure 7: Autoimmunology disease rule

- *nr księgi głównej 12354*
  'nr of the main document 12354'

- *Nr. księgi głównej 13578*
  'Nr. of the main document 13578'

The grammar rules recognising the important information are often relatively simple. There is no need to use any morphological features in rules, because we do not have to precisely delimit phrases. Searched (key) phrases consist very often of words which are very strongly connected with particular notions. For example, if we find a phrase *stopy cukrzycowej* 'diabetic foot', it is practically certain that it concerns a complication. Only base word forms (values of STEM attribute) from the morphological analyser output turned out to be necessary here.

Fig. 7 contains a simple rule recognising autoimmunology diseases. It seeks for any occurrence of the following pattern: <disease, autoimmunology-disease-specification>. The first line of the rule recognises a word: *choroba* or *zespół* e.g., 'disease'. The second line requires an entry from the domain dictionary which represents an autoimmunology disease. Its type (variable `#type`) is unified with the value of the attribute AUTOIMM_DISEASE in the output structure.

3.3 Difficult Issues

Although the results of the program are quite good, there are some difficult issues which cause errors.

[2] Abbreviations are not present in the morphological dictionary.

We have to cope with negation, which sometimes is difficult to determine. In the following phrase: *bez obecności retinopatii* ‘without presence of retinopathy’, it is not enough to identify the keyword *retinopatii* ‘retinopathy’, it is necessary to recognise negation expressed in the form of the negative preposition *bez* ‘without’. Here, the negation appeared just before the keyword, and it can be easily noticed, but sometimes a negation is far from a keyword, and is difficult to process with shallow parsing methods. Let us consider the following sentence: *Nie stwierdzono późnych powikłań cukrzycy o typie mikroangiopatii.* ‘there were no long-lasting diabetes complications of microagiopathy type ’. In this case, the negation *nie stwierdzono* ‘there were no’ is at the beginning of the sentence and the keyword *mikroangiopatii* ‘microangiopathy’ is the last word of the sentence. The above phrase is recognised with the rule in Fig. 8. It refers to the base forms of certain words and to the domain lexicon in order to identify a complication (variable #t). The same rule recognise, among other, the following phrases which meaning is the same as the previous one.

- *nie wykryto obecności późnych powikłań cukrzycowych pod postacią mikroangiopatii*,
- *nie występują późne powikłania cukrzycowe o charakterze mikroangiopatii*,
- *Nie stwierdzono późnych zmian cukrzycowych w postaci mikroangiopatii.*

In the very similar example: *Nie stwierdzono późnych powikłań cukrzycy z wyjątkiem mikroangiopatii.* ‘there were no long-lasting diabetes complications excluding microagiopathy’ the case is just the opposite, and the *microangiopathy* should be recognised. So, to properly identify whether a patient has or hasn’t microangiopathy we have to analyse the whole sentence.

Some problems are caused by keywords which have different interpretation depending on the context. e.g., *mikroalbuminuria* refers to a complication in the phrase *wystąpiła mikroalbuminuria* ‘microalbuminuria appeared’ and denotes a test in the phrase *Mikroalbuminuria: 25 mg/dobę* ‘Microalbuminuria: 25 mg/day’. In this case we determine the meaning of an ambiguous notion according to its context.

```
brak_powiklan :>
   morph & [STEM “nie”]                              ;; ‘no’
   (morph & [STEM “stwierdzić”] |                    ;; ‘recognise’
            morph & [STEM “wystepować”] |
            morph & [STEM “wykryć”])
   (morph & [STEM “obecność”])?
   morph & [STEM “późny”]                            ;; ‘long-lasting’
   (morph & [STEM “powikłanie”] |                    ;; ‘complication’
            morph & [STEM “zmiana”])
   (morph & [STEM “cukrzycowy”] |                    ;; ‘diabetes’
            morph & [STEM “cukrzyca”])
   (morph & [STEM “w”] |                             ;; preposition
            morph & [STEM “pod”] | morph & [STEM “o”])
   (morph & [STEM “postać”] |                        ;; ‘type’
            morph & [STEM “typ”] | morph & [STEM “charakter”])
   gazetteer & [GTYPE gaz_comp, G_CONCEPT #t]
->no_comp_str & [N_COMP #t].
```

Figure 8: The rule recognising the lack of a specified complication

The next thing that should be taken into account, is that sometimes several pieces of information have to be recognised with one rule. In the following coordinated phrase: *retinopatię prostą oka lewego oraz proliferacyjna oka prawego z makulopatią w obu oczach* ‘nonproliferative rethinopathy in the left eye and proliferative (rethinopathy) in the right eye with maculopathy in both eyes’ we have to recognise both types of rethinopathy with maculopathy and create a list of complications as the output structure, see Fig. 9. The rule almost entirely refers to notions from the domain dictionary. It identifies a combination of notions denoting retinopathy. The domain dictionary contains both Polish and Latin (words in this case both languages are used by doctors) referring to this complication.

In order to recognise precisely given information, one tends to write complex rules describing entire phrases instead of separated terms. The crucial problem for the effectiveness of complex IE rules is that Polish is a free word language. This greatly increases the number of ways the same idea can be expressed. Let us consider the following examples:

- *Wieloletnia, niekontrolowana cukrzyca typu 2*, long-lasting uncontrolled diabetes type 2,
- *Niekontrolowana, wieloletnia cukrzyca typu 2*,
- *Wieloletnia cukrzyca typu 2, niekontrolowana*,
- *Cukrzyca wieloletnia typu 2, niekontrolowana*.

```
retino_koord1:>
  gazetteer & [GTYPE gaz_comp, G_CONCEPT retinopathy_t]
  token ?
  gazetteer & [GTYPE gaz_retino, G_CONCEPT #r1]
  (token){0,2}
  (token & [SURFACE “i”] | token & [SURFACE “oraz”] |
       token &[SURFACE “et”] | token & [TYPE comma])
  (gazetteer &
       [GTYPE gaz_comp, G_CONCEPT retinopathy_t])?
  token ?
  gazetteer & [GTYPE gaz_retino, G_CONCEPT #r2]
  (token){0,2}
  ((token & [SURFACE “z”] | token & [SURFACE “cum”] |
       token & [SURFACE “i”] | token & [SURFACE “oraz”])
  gazetteer & [GTYPE gaz_macul, G_CONCEPT yes & #z1 ])?
->
comp_l_str & [ COMP_L complication_list &
  [FIRST retinopathy_str &[ RETINOPATHY_T #r1 ,
         WITH_MACULOPATHY #z1 ],
  REST complication_list &
       [FIRST retinopathy_str &[ RETINOPATHY_T #r2 ,
         WITH_MACULOPATHY #z1 ],
       REST *null* ]]].
```

Figure 9: Retinopathy coordination rule

All phrases mean: ‘Long-lasting, uncontrolled, type 2 diabetes’. Every word of these phrases carries important information: *wieloletni* ‘long-lasting’, *niekontrolowany* ‘uncontrolled’, *typ 2* ‘type 2’. But they should be identified as important only in context of the keyword *cukrzyca* ‘diabetes’. The only solution is to recognise the whole phrase through one rule. So, we ought to predict all possible configurations of words and write a lot of rules that identify subsequent permutations of keywords, which might be difficult. Thus, some omissions of information can be caused by insufficient coverage by grammar rules (see sec. 4).

The information we searched for can be divided into two types. Many facts were originally written in the documents in a standardised way, for example the value of the BMI parameter, or phrases describing complications. For these parts of information, the probability of error is rather small and is related mostly to the occurrence of complicated negation or coordination. But some of the features can be expressed in many ways. In this case, the program recall can depend on the particular physicians’ writing styles. An example is the information about continuation of diabetes treatment. In this case we have to identify information about continuation of a treatment (can be expressed in many ways) in the context of a phrases denoting diabetes. This context is important because, in the texts, there are sometimes phrases describing continuation of treatment of not diabetes but other illness. A few samples are given below:

- *Kontynuowano leczenie cukrzycy dotychczasowym systemem wielokrotnych wstrzyknięć* ‘The diabetes treatment was continued on the same basis of multiple injections’,

- *Utrzymano dotychczasowy system wielokrotnych wstrzyknięć insuliny* ‘The current system of multiple insulin injections has been maintained’,

- *Kontynuowano dotychczasowy schemat leczenia cukrzycy* ‘The current schema of diabetes treatment was continued’,

- *Kontynuowano dotychczasowe leczenie hipotensyjne* ‘The current treatment of hypotension was continued’ — this phrase is not about diabetes!

A fact that a patient was educated for diabetes is another example of information which can be expressed in many ways. Any phrase indicating that a patient was informed or taught about something or something was discussed with a patient is interpreted as the information about education. We are not interested in details of education but still we have to recognise 13 different constructions describing education.

- *Omówiono z chorym zasady diety, samokontroli i adaptacji dawek insuliny* ‘Diet, self-control and adaptation of insulin doses were discussed with the patient’,

- *Nauczono chorego posługiwać sie pompą insulinową i glukometrem.* ‘The patient was taught how to use an insulin pump and a glucometer.’,

- *W czasie pobytu w Klinice prowadzono edukację chorej dotyczącą cukrzycy.* ‘During the patient stay in the Clinic, the patient was educated for diabetes.’,

- *Po odbyciu szkolenia z zakresu podstawowych wiadomości o cukrzycy wypisano chorą...* 'After learning the basic information about diabetes, the patient was discharged...'.

4 IE results evaluation

Part of the data was used as a training set, the evaluation was made on the other 50 previously unseen reports. From above 60 attributes, the partial evaluation concerned only 7. The evaluated attributes are of different type: retinopathy is a keyword but we still deal with the problem of negation and coordination. Words denoting uncontrolled diabetes can refer not only to diabetes so they should be recognized only in specific contexts. Attributes: education and diet modification are represented in the texts by complex phrases.

Results are presented in Fig. 10. The worst results were observed for diabetes balance recognition. It was due to the fact that keywords representing this information had to be recognised in the context of the word *cukrzyca* 'diabetes', (see 3.3) and sometimes the distance between these words is too far. 4 occurrences of wrongly recognised *retinopathy* were caused by the unpredicted negated phrases.

	phrases	precision	recall
uncontrolled diabetes	61	100	68,85
retinopathy (total)	50	92,5	98
nonproliferative	35	100	100
preproliferative	9	100	88,89
proliferative	5	100	100
unspecified	1	20	100
diabetic education	19	100	94,74
diet modification	1	100	100

Figure 10: IE evaluation of 50 reports

5 Database Organization

The data obtained from the IE system is a huge XML file. The attribute values included within it were subsequently introduced into a relational database which can be searched and analysed. At the database filling stage some additional postprocessing of data was done. This concerned, among others, the following problems:

- detection and omission of information of patient not suffering from diabetes,
- detection and omission of not complete data (reports not sufficiently filled up with data),
- omission of redundant data and choosing the most detailed information (e.g. about types of complications)
- selecting highest levels for blood test results.

The database consists of 20 tables containing all extracted information about a patient, his/her illness and the recommended treatment. At the moment, the database contains 388 hospitalisation descriptions of 387 patients. 254 cases were qualified as diabetes type 2, 129 as type 1 and 5 as type other. 556 complications for 256 patients and 304 insulin treatment schemas have been recognised.

6 System Overview and Evaluation

The main aim of the work was creation of a system that processes diabetic patients' hospital documentation automatically and inserts the extracted data into a relational database. The database can be searched for using SQL queries or a specialized program dedicated for doctors which enables queries by example. The system architecture is given in Fig. 11. The processing procedure consisted of four stages:

- text preprocessing including format unification and data anonymization (Perl scripts),
- information extraction based on the domain model (Protégé), Polish morphological lexicon (Morfeusz) and the domain lexicon,
- postprocessing: data cleaning and structuring (Perl scripts),
- insertion data into a relational database (Postgres).

The evaluation of the system was done simultaneously with IE evaluation on the same set of 50 reports. The results are presented in Fig. 12. The final recognition of the uncontrolled diabetes was higher due to repetition of the same information in one document.

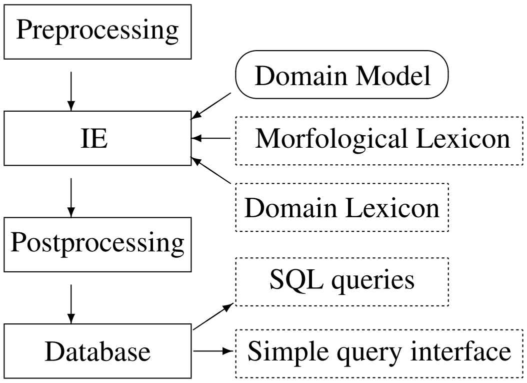


Figure 11: System architecture

	cases	precision	recall
uncontolled diabetes	37	100	86,49
retinopathy (total)	22	88	100
nonproliferative	14	100	100
preproliferative	4	100	88,89
proliferative	3	100	100
unspecified	1	25	100
diabetic education	19	100	94,74
diet modification	1	100	100

Figure 12: Overall system evaluation of 50 reports

7 Conclusions

For the chosen domain, the rule-based IE method seems to be the best one. Learning techniques are hard to apply due to: a great number of attributes searched for (in comparison to the amount of available texts) and their inter connections and crucial dependence on negation and coordination occurrences. Good precision and recall values make this method practically usable for information extraction from free patients' documentation. We plane to use our tools for creating annotated corpora (manually corrected) which are necessary for training statistical models.

Of course the portability of the method is poor. The grammars written for a particular domain can be developed to cover more facts and details but their extendibility to another domain is problematic.

Acknowledgment

This work was done in cooperation with Bródnowski Hospital in Warsaw and was partially financed by the Polish national project number 3 T11C 007 27.

References

Razvan Bunescu, Ruifang Ge, Rohit. J. Kate, Raymond J. Mooney, and Yuk Wah Wong. 2003. Learning to extract proteins and their interactions from Medline abstracts, *Proceedings of ICML-2003 Workshop on Machine Learning in Bioinformatics*, pp. 46-53, Washington, DC.

Witold Drożdżyński, Hans-Ulrich Krieger, Jakub Piskorski, Ulrich Schäfer and Feiyu Xu. 2004. Shallow Processing with Unification and Typed Feature Structures – Foundations and Applications. *German AI Journal KI-Zeitschrift, 01/04*.

Udo Hahn, Martin Romacker and Stefan Schulz. 2002. Creating knowledge repositories from biomedical reports: The MEDSYNDIKATE text mining system. In *Proceedings PSB 2002*, pages 338–349.

Henk Harkema, Andrea Stzer, Rob Gaizauskas, Mark Hepple, Richard Power and Jeremy Rogers. 2005. Mining and Modelling Temporal Clinical Data. In: Proceedings of the UK e-Science All Hands Meeting 2005, Nottingham UK.

George Hripcsak, John Austin, Philip O. Alderson and Carol Friedman, 2002. Use of Natural Language Processing to Translate Clinical Information from a Database of 889,921 Chest Radiographic Reports *Radiology*.

Agnieszka Mykowiecka, Anna Kupść, Małgorzata Marciniak, 2005. Rule-based Medical Content Extraction and Classification, *Proceedings of ISMIS 2005, Springer-Verlag*.

Jakub Piskorski, Peter Homola, Małgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiórkowski and Marcin Woliński. 2004. Information Extraction for Polish using the SProUT Platform. In: *Proceedings of ISMIS 2004, Zakopane*, pp. 225–236, Springer-Verlag.

Amund Tveit and Rune Saetre, 2005. ProtChew: Automatic Extraction of Protein Names from Biomedical Literature, *Proceedings of the 21st International Conference on Data Engineering Workshops.*

Marcin Woliński. 2006. Morfeusz – a Practical Tool for the Morphological Analysis of Polish. *Procceedings of IIS: IIPWM´06*. Advances in Soft Computing, Springer-Verlag

Roman Yangarber, Winston Lin and Ralph Grishman. 2002. Unsupervised Learning of Generalized Names. *Proceedings of the 19th International Conference on Computational Linguistics, COLING 2002.*

Pierre Zweigenbaum (ed.). 1994. MENELAS: An Access System for Medical Records Using Natural Language, In: *Computer Methods and Programs in Biomedicine* vol. 45.

Towards the Automatic Extraction of Definitions in Slavic

[1]**Adam Przepiórkowski**
[2]**Łukasz Degórski**
[8]**Beata Wójtowicz**
Institute of Computer Science PAS
Ordona 21, Warsaw, Poland
adamp@ipipan.waw.pl
ldegorski@bach.ipipan.waw.pl
beataw@bach.ipipan.waw.pl

[3]**Miroslav Spousta**
[7]**Vladislav Kuboň**
Charles University
Malostranské náměstí 25
Prague, Czech Republic
spousta@ufal.ms.mff.cuni.cz
vk@ufal.ms.mff.cuni.cz

[4]**Kiril Simov**
[5]**Petya Osenova**
Institute for Parallel Processing BAS
Bonchev St. 25A, Sofia, Bulgaria
kivs@bultreebank.org
petya@bultreebank.org

[6]**Lothar Lemnitzer**
University of Tübingen
Wilhelmstr. 19, Tübingen, Germany
lothar@sfs.uni-tuebingen.de

Abstract

This paper presents the results of the preliminary experiments in the automatic extraction of definitions (for semi-automatic glossary construction) from usually unstructured or only weakly structured e-learning texts in Bulgarian, Czech and Polish. The extraction is performed by regular grammars over XML-encoded morphosyntactically-annotated documents. The results are less than satisfying and we claim that the reason for that is the intrinsic difficulty of the task, as measured by the low interannotator agreement, which calls for more sophisticated deeper linguistic processing, as well as for the use of machine learning classification techniques.

1 Introduction

The aim of this paper is to report on the preliminary results of a subtask of the European Project *Language Technology for eLearning* (http://www.lt4el.eu/) consisting in the identification of term definitions in eLearning materials (Learning Objects; henceforth: LOs), where definitions are understood pragmatically, as those text fragments which may, after perhaps some minor editing, be put into a glossary. Such automatically extracted term definitions are to be presented to the author or the maintainer of the LO and, thus, significantly facilitate and accelerate the creation of a glossary for a given LO. From this specification of the task it follows that good recall is much more important than good precision, as it is easier to reject wrong glossary candidates than to browse the LO for term definitions which were not automatically spotted.

The project involves 9 European languages including 3 Slavic (and, regrettably, no Baltic) languages: one South Slavic, i.e., Bulgarian, and two West Slavic, i.e., Czech and Polish. For all languages, shallow grammars identifying definitions have been constructed; after mentioning some previous work on Information Extraction (IE) for Slavic languages and on extraction of definitions in section 2, we briefly describe the three Slavic grammars developed within this project in section 3. Section 4 presents the results of the application of these grammars to LOs in respective languages. These results are evaluated in section 5, where main problems, as well as some possible solutions, are discussed. Finally, section 6 concludes the paper.

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 43–50,
Prague, June 2007.

2 Related Work

Definition extraction is an important NLP task, most frequently a subtask of terminology extraction (Pearson, 1996), the automatic creation of glossaries (Klavans and Muresan, 2000; Klavans and Muresan, 2001), question answering (Miliaraki and Androutsopoulos, 2004; Fahmi and Bouma, 2006), learning lexical semantic relations (Malaisé et al., 2004; Storrer and Wellinghoff, 2006) and automatic construction of ontologies (Walter and Pinkal, 2006). Tools for definition extraction are invariably language-specific and involve shallow or deep processing, with most work done for English (Pearson, 1996; Klavans and Muresan, 2000; Klavans and Muresan, 2001) and other Germanic languages (Fahmi and Bouma, 2006; Storrer and Wellinghoff, 2006; Walter and Pinkal, 2006), as well as French (Malaisé et al., 2004). To the best of our knowledge, no previous attempts at definition extraction have been made for Slavic, with the exception of some work on Bulgarian (Tanev, 2004; Simov and Osenova, 2005).

Other work on Slavic information extraction has been carried out mainly for the last 5 years. Probably the first forum where such work was comprehensively presented was the International Workshop on Information Extraction for Slavonic and Other Central and Eastern European Languages (IESL), RANLP, Borovets, 2003, Bulgaria. One of the papers presented there, (Drożdżyński et al., 2003), discusses shallow SProUT (Becker et al., 2002) grammars for Czech, Polish and Lithuanian. SProUT has subsequently been extensively used for the information extraction from Polish medical texts (Piskorski et al., 2004; Marciniak et al., 2005).[1]

3 Shallow Grammars for Definition Extraction

The input to the task of definition extraction is XML-encoded morphosyntactically-annotated text, possibly with some keywords already marked by an independent process. For example, the representation of a Polish sentence starting as *Konstruktywizm kładzie nacisk na* (Eng. “Constructivism puts emphasis on”) may be as follows:[2]

```
<s id="s9">
<markedTerm id="mt7" kw="y">
<tok base="konstruktywizm" ctag="subst"
 id="t253"
 msd="sg:nom:m3">Konstruktywizm</tok>
</markedTerm>
<tok base="klasc" ctag="fin" id="t254"
 msd="sg:ter:imperf">kladzie</tok>
<tok base="nacisk" ctag="subst" id="t255"
 msd="sg:acc:m3">nacisk</tok>
<tok base="na" ctag="prep" id="t256"
 msd="acc">na</tok>
[...]
<tok base="." ctag="interp" id="t273">.
</tok>
</s>
```

For each language, definitions were manually marked in two batches of texts: the first batch, consulted during the process of grammar development, contained at least 300 definitions, and the second batch, held out for evaluation, contained about 150 definitions. All grammars are regular grammars implemented with the use of the `lxtransduce` tool (Tobin, 2005), a component of the LTXML2 toolset developed at the University of Edinburgh.[3] An example of a simple rule for prepositional phrases is given below:

```
<rule name="PP">
 <seq>
  <query match="tok[@ctag = ’prep’]"/>
  <ref name="NP1">
   <with-param name="case" value="’’"/>
  </ref>
 </seq>
</rule>
```

This rule identifies a sequence whose first element is a token tagged as a preposition and whose subsequent elements are identified by a rule called `NP1`. This latter rule (not shown here for brevity) is a parameterised rule which finds a nominal phrase of a given case, but the way it is called above ensures that it will find an NP of any case.

[1] SProUT has not been seriously considered for the task at hand for two reasons: first, it was decided that only open source tools will be used in the current project, if only available, second, the input format to the current task is morphosyntactically-annotated XML-encoded text, rather than raw text, as normally expected by SProUT. The second obstacle could be removed by converting input texts to the SProUT-internal XML representation.

[2] Part of the representation has been replaced by ‘[...]’.

[3] Among the tools considered here were also CLaRK (Simov et al., 2001), ultimately rejected because it currently does not work in batch mode, and GATE / JAPE (Cunningham et al., 2002), not used here because we found GATE’s handling of previously XML-annotated texts rather cumbersome and ill-documented. Cf. also fn. 1.

Currently the grammars show varying degrees of sophistication, with a small Bulgarian grammar (8 rules in a 2.5-kilobyte file), a larger Polish grammar (34 rules in a 11 KiB file) and a sophisticated Czech grammar most developed (147 rules in a 28 KiB file). The patterns defined by these three grammars are similar, but sufficiently different to defy an attempt to write a single parameterised grammar.[4] The remainder of this section briefly describes the grammars.

3.1 Bulgarian

The Bulgarian grammar is manually constructed after examination of the manually annotated definitions. Here is a list of the rule schemata, together with the number and percentage of matching definitions:

Pattern	#	%
NP is NP	140	34.2
NP verb NP	18	29.8
NP - NP	21	5.0
This is NP	15	3.7
It represents NP	4	1.0
other patterns	107	26.2

Table 1: Bulgarian definition types

In the second schema above, "verb" is a verb or a verb phrase (not necessarily a constituent) which is one of the following: 'представлява' (to represent), 'показва' (to show), 'означава' (to mean), 'описва' (to describe), 'се използва' (to be used), 'позволява' (to allow), 'дава възможност да' (to give opportunity), 'се нарича' (is called), 'подобрява' (to improve), 'осигурява' (to ensure), 'служи за' (to serve as), 'се разбира' (to be understood as), 'обозначава' (to denote), 'съдържа' (to contain), 'определя' (to determine), 'включва' (to include), 'се дефинира като' (is defined as), 'се основава на' (is based on).

We classify the rules in five types: copula definitions, copula definitions with anaphoric relation, copula definitions with ellipsis of the copula, definitions with a verb phrase, definitions with a verb phrase and anaphoric relation. Each of these types of definitions defines an NP (sometimes via anaphoric relation) by another one. There are some variations of the models where some parenthetical expressions are presented in the definition.

The grammar contains several most important rules for each type. The different verb patterns are encoded as a lexicon. For some of the rules, variants with parenthetical phrases are also encoded. The rest of the grammar is devoted to the recognition of noun phrases and parenthetical phrases. For parenthetical phrases, we have encoded a list of such possible phrases, extracted on the basis of a bigger corpus. The NP grammar in our view is the crucial grammar for recognition of the definitions. Most work now has to be invested into developing the more complex and recursive NPs.

3.2 Czech

The Czech grammar for definition context extraction is constructed to follow both linguistic intuition and observation of common patterns in manually annotated data.

We adapted a grammar[5] based mainly on the observation of Czech Wikipedia entries. Encyclopedia definitions are usually clear and very well structured, but it is quite difficult to find such well-formed definitions in common texts, including learning objects. The rules were extended using part of our manually annotated texts, evaluated and adjusted in several iterations, based on the observation of the annotated data.

Pattern	#	%
NP is/are NP	52	21.2
NP verb NP	45	18.4
structural	39	15.9
NP (NP)	30	12.2
NP -/:/= NP	20	8.2
other patterns	59	24.1

Table 2: Czech definition types

There are 21 top level rules, divided into five categories. Most of the correctly marked definitions fall into the copula verb ('is/are') category. The sec-

[4] Because of this relative language-dependence of definition patters, which includes, e.g., idiosyncratic case information, we have not seriously considered re-using rules for other, non-Slavic, languages.

[5] The grammar was originally developed by Nguyen Thu Trang.

ond most successful rule is the one using selected verbs like ‘definuje’ (defines), ‘znamená’ (means), ‘vymezuje’ (delimits), ‘představuje’ (presents) and several others. The remaining categories make use of the typical patterns of characters (dash, colon, equal sign and brackets) or additional structural information (e.g., HTML tags).

3.3 Polish

The Polish grammar rules are divided into three layers. Similarly to the Czech grammar, each layer only refers to itself or lower layers. This allows for expressing top level rules in a clear and easily manageable way.

The top level layer consists of rules representing typical patterns found in Polish documents:

Pattern	#	%
NP (...) are/is NP-INS	40	15.6
NP -/: NP	39	15.2
NP (are/is) *to* NP-NOM	27	10.6
NP VP-3PERS	25	9.8
NP - i.e./or WH-question	11	4.3
N ADJ - PPAS	8	3.1
NP, i.e./or NP	7	2.7
NP-ACC one may describe/define as NP-ACC	5	2.0
other patterns (not in the grammar)	94	36.7

Table 3: Polish definition types

The middle layer consists of rules catching patterns such as “simple NP in given case, followed by a sequence of non-punctuation elements” or “copula”.

The bottom layer rules basically only refer to POS markup in the input files (or other bottom layer rules).

4 Results

As mentioned above, the testing corpus for each language consists of about 150 definitions, unseen during the construction of the grammar.[6]

The Bulgarian test corpus, containing around 76,800 tokens, consists of the third part of the Calimera guidelines (`http://www.calimera.org/`). We view this document as appropriate for testing because it reflects the chosen domain and it combines definitions from otherwise different subdomains, such as XML language, Internet usage, etc. There are 203 manually annotated definitions in this corpus: 129 definitions contained in one sentence, 69 definitions split across 2 sentences, 4 definitions in 3 sentences and one definition in 4 sentences. Note that the real test part is the set of the 129 definitions in one sentence, since the Bulgarian grammar does not consider cross-sentence definitions in any way.

Czech data used for evaluation consist of several chapters of the Calimera guidelines and Microsoft Excel tutorial. The tutorial is a typical text used in e-learning, consisting of five chapters describing sheets, tables, formating, graphs and lists. The corpus consists of over 90,000 tokens and contains 162 definitions, out of which 153 are contained in a single sentence, 6 span 2 sentences, and 3 definitions span 3 sentences.

Polish test corpus consists of over 83,200 tokens containing 157 definitions: 148 definitions are contained within one sentence, while 9 span 2 sentences. The corpus is made up of 10 chapters of a popular introduction to and history of computer science and computer hardware.

Each grammar was quantitatively evaluated by comparing manually annotated files with the same files annotated automatically by the grammar. After considering various ways of quantitative evaluation, we decided to do the comparison at token level: precision was calculated as the ratio of the number of those tokens which were parts of *both* a manually marked definition and an automatically discovered definition to the number of all tokens in automatically discovered definitions, while recall was taken to be the ratio of the number of tokens simultaneously in both kinds of definitions to the number of tokens in all manually annotated definitions. Since, for this task, recall is more important than precision, we used the F_2-measure for the combined result.[7]

[6] Obviously, three different corpora had to be used to evaluate the grammars for the three languages, but the corpora are similar in size and character, so any differences in results stem mostly from the differences in the three grammars.

[7] In general, $F_\alpha = (1+\alpha) \cdot (\text{precision} \cdot \text{recall})/(\alpha \cdot \text{precision} + \text{recall})$. Perhaps α larger than 2 could be used, but it is currently not clear to us what criteria should be assumed when

The results for the three grammars are given in Table 4. Note that the processing model for Czech differs from the other two languages, as the input text is converted to a flat format, as described in section 5.3, and grammar rules are sensitive to sentence boundaries (and may operate over them).

	precision	recall	F_2
Bulgarian	20.5%	2.2%	3.1
Czech	18.3%	40.7%	28.9
Polish	14.8%	22.2%	19.0

Table 4: Token-based evaluation of shallow grammars

5 Evaluation and Possible Improvements

5.1 Interannotator Agreement

We calculated Cohen's kappa statistic (1) for the current task, where both the relative observed agreement among raters $\Pr(a)$ and the probability that agreement is due to chance $\Pr(e)$ where calculated at token level.

$$\kappa = \frac{\Pr(a) - \Pr(e)}{1 - \Pr(e)} \tag{1}$$

More specifically, we assumed that two annotators agree on a token if the token belongs to a definition either according to both annotations or according to neither. In order to estimate the probability of agreement due to chance $\Pr(e)$, we measured, separately for each annotator, the proportion of tokens found in definitions to all tokens in text, which resulted in two probability estimates p_1 and p_2, and treated $\Pr(e)$ as the probability that the two annotators agree if they randomly, with their own probability, classify a token as belonging to a definition, i.e.:

$$\Pr(e) = p_1 \cdot p_2 + (1 - p_1) \cdot (1 - p_2) \tag{2}$$

The interannotator agreement (IAA) was measured this way for Czech and Polish, where — for each language — the respective test corpus was annotated by two annotators. The results are 0.44 for Czech and 0.31 for Polish. Such results are very low for any classification task, and especially low for a binary classification task. They show that the task of identifying definitions in running texts and agreeing on which parts of text count as a definition is intrinsically very difficult. They also call for the reconsideration of the evaluation and IAA measurement methodology based on token classification.[8]

5.2 Evaluation Methodology

To the best of our knowledge, there is no established evaluation methodology for the task of definition extraction, where definitions may span several sentences.[9] For this reason we evaluated the results again, in a different way: we treated an automatically discovered definition as correct, if it overlapped with a manually annotated definition. We calculated precision as the number of automatic definitions overlapping with manual definitions, divided by the number of automatic definitions, while recall — as the number of manual definitions overlapping automatic definitions, divided by the number of manual definitions.[10]

The results for the three grammars, given in Table 5, are much higher than those in Table 4 above, although still less than satisfactory.

	precision	recall	F_2
Bulgarian	22.5%	8.9%	11.1
Czech	22.3%	46%	33.9
Polish	23.3%	32%	28.4

Table 5: Definition-based evaluation of shallow grammars

5.3 Definitions and Sentence Boundaries

Regardless of the inherent difficulties of the task and difficulties with the evaluation of the results, there is clear room for improvement; one possible path

deciding on the exact value of α. Note that it would not make sense to use recall alone, as it is trivial to write all-accepting grammars with 100% recall.

[8] A better approximation would be to measure IAA on the basis of sentence or (as suggested by an anonymous reviewer) NP classification; we intend to pursue this idea in future work.

[9] With the assumption that definitions are no longer than a sentence, usually the task is treated as a classification task, where sentences are classified as definitional or not, and appropriate precision and recall measures are applied at sentence level.

[10] At this stage definition fragments distributed across a number of different sentences were treated as different definitions, which negatively affects the evaluation of the Bulgarian grammar, as the Bulgarian test corpus contains a large number of multi-sentence definitions.

to explore concerns multi-sentence definitions. As noted above, for all languages considered here, there were definitions which were spanning 2 or more sentences; this turned out to be a problem especially for Bulgarian, were 36% of definitions crossed a sentence boundary.[11]

Such multi-sentence definitions are a problem because in the DTD adopted in this project definitions are subelements of sentences rather than the other way round. In case of a multi-sentence definition, for each sentence there is a separate element encapsulating the part of the definition contained in this sentence. Although these are linked via special attributes and the information that they are part of the same definition can subsequently be recovered, it is difficult to construct an `lxtransduce` grammar which would be able to automatically mark such multi-sentence definitions: an `lxtransduce` grammar expects to find a sequence of elements and wrap them in a single larger element.

A solution to this technical problem has been implemented in the Czech grammar, where first the input text is flattened (via an XSLT script), so that, e.g.:

```
<par id="d1p2">
 <s id="d1p2s1">
  <tok id="d1p2s1t1" base="Pavel"
    ctag="N" msd="NMS1-----A----">
    Pavel</tok>
  <tok id="d1p2s1t2" base="satrapa"
   ctag="N" msd="NMS1-----A----">
   Satrapa</tok>
 </s>
</par>
```

becomes:

```
<par id="Sd1p2"/>
<s id="Sd1p2s1"/>
<tok id="d1p2s1t1" base="Pavel"
 ctag="N" msd="NMS1-----A----">
 Pavel</tok>
<tok id="d1p2s1t2" base="satrapa"
 ctag="N" msd="NMS1-----A----">
 Satrapa</tok>
<s id="Ed1p2s1"/>
<par id="Ed1p2"/>
```

This flattened representation is an input to a grammar which is sensitive to the empty `s` and `par` elements and may discover definitions containing such elements; in such a case, the postprocessing script, which restores the hierarchical paragraph and sentence structure, splits such definitions into smaller elements, fully contained in respective sentences.

5.4 Problems Specific to Slavic

At least in case of the two West Slavic languages considered here, the task of writing a definition grammar is intrinsically more difficult than for Germanic or Romance languages, mainly for the following two reasons.

First, Czech and Polish have very rich nominal inflection with a large number of paradigm-internal syncretisms. These syncretisms are a common cause of tagger errors, which percolate to further stages of processing. Moreover, the number of cases makes it more difficult to encode patterns like “NP verb NP”, as different verbs may combine with NPs of different case. In fact, even two different copulas in Polish take different cases!

Second, the relatively free word order increases the number of rules that must be encoded, and makes the grammar writing task more labour-intensive and error-prone. The current version of the Polish grammar, with 34 rules, is rather basic, and even the 147 rules of the Czech grammar do not take into consideration all possible patterns of grammar definitions. As Tables 4 and 5 show, there is a positive correlation between the grammar size and the value of F_2, and the Bulgarian and Polish grammars certainly have room to grow. Moreover, a path that is well worth exploring is to drastically increase the number of rules and, hence, the recall, and then deal with precision via Machine Learning methods (cf. section 5.6).

5.5 Levels of Linguistic Processing

The work reported here has been an excercise in definition extraction using shallow parsing methods. However, the poor results suggest that this is one of the tasks that require a much more sophisticated and deeper approach to language analysis. In fact, in turns out that virtually all successful attempts at definition extraction that we are aware of build on worked-out deep linguistic approaches (Klavans and

[11] An example of a Polish manually annotated multi-sentence definition is: *. . . opracowano techniki antyspamowe. Techniki te drastycznie zaniżają wartość strony albo ją banują. . .* (Eng. “. . . anti-spam techniques were developed. Such techniques drastically lower the value of the page or they ban it. . . ”). The definition is split into two fragments fully contained in respective sentences: *techniki antyspamowe* and *Techniki te. . . .* No attempt at anaphora resolution is made.

Muresan, 2000; Fahmi and Bouma, 2006; Walter and Pinkal, 2006), some of them combining syntactic and semantic information (Miliaraki and Androutsopoulos, 2004; Walter and Pinkal, 2006).

Unfortunately, for most Baltic and Slavic languages, such deep parsers are unavailable or have not yet been extensively tested on real texts. One exception is Czech, where a number of parsers were already described and evaluated (on the Prague Dependency Treebank) in (Zeman, 2004, § 14.2); the best of these parsers reach 80–85% accuracy.

For Polish, apart from a number of linguistically motivated toy parsers, there is a possibly wide coverage deep parser (Woliński, 2004), but it has not yet been evaluated on naturally occurring texts. The situation is probably most dire for Bulgarian, although there have been attempts at the induction of a dependency parser from the BulTreeBank (Marinov and Nivre, 2005; Chanev et al., 2006).

Nevertheless, if other possible paths of improvement suggested in this section do not bring satisfactory results, we plan to make an attempt at adapting these parsers to the task at hand.

5.6 Postprocessing: Machine Learning and Keyword Identification

Various approaches to the machine learning treatment of the task of classifying sentences or snippets as definitions or non-definitions can be found, e.g., in (Miliaraki and Androutsopoulos, 2004; Fahmi and Bouma, 2006) and references therein. In the context of the present work, such methods may be used to postprocess apparent definitions found at earlier processing stages and decide which of them are genuine definitions. For example, (Fahmi and Bouma, 2006) report that a system trained on 2299 sentences, including 1366 definition sentences, may increase the accuracy of a definition extraction tool from 59% to around 90%.[12]

Another possible improvement may consist in, again, aiming at very high recall and then using an independent keyword detector to mark keywords (and key phrases) in text and classifying as genuine definitions those definitions, whose defined term has been marked as a keyword.

Whatever postprocessing technique or combination of techniques proves most efficient, it seems that the linguistic processing should aim at high recall rather than high precision, which further justifies the use of the F_2 measure for evaluation.[13]

6 Conclusion

To the best of our knowledge, this paper is the first report on the task of definition extraction for a number of Slavic languages. It shows that the task is intrinsically very difficult, which partially explains the relatively low results obtained. It also calls attention to the fact that there is no established evaluation methodology where possibly multi-sentence definitions are involved and suggests what such methodology could amount to. Finally, the paper suggests ways of improving the results, which we hope to follow and report in the future.

References

Markus Becker et al. 2002. SProUT — shallow processing with typed feature structures and unification. In *Proceedings of the International Conference on NLP (ICON 2002)*, Mumbai, India.

Sharon A. Caraballo. 2001. *Automatic Construction of a Hypernym-Labeled Noun Hierarchy from Text*. Ph. D. dissertation, Brown University.

Atanas Chanev, Kiril Simov, Petya Osenova, and Svetoslav Marinov. 2006. Dependency conversion and parsing of the BulTreeBank. In *proceedings of the LREC workshop Merging and Layering Linguistic Information*, Genoa, Italy.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A framework and graphical development environment for robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics*.

Witold Drożdżyński, Petr Homola, Jakub Piskorski, and Vytautas Zinkevičius. 2003. Adapting SProUT to processing Baltic and Slavonic languages. In *Information Extraction for Slavonic and Other Central and Eastern European Languages*, pp. 18–25.

Ismail Fahmi and Gosse Bouma. 2006. Learning to identify definitions using syntactic features. In *Proceedings of the EACL 2006 workshop on Learning Structured Information in Natural Language Applications*.

[12] The numbers are so high “probably due to the fact that the current corpus consists of encyclopedic material only” (Fahmi and Bouma, 2006, fn. 4).

[13] Note that the situation here is different than in the task of acquiring hyponymic relations from texts, where high-precision manual rules (Hearst, 1992) must be augmented with statistical clustering methods to increase recall (Caraballo, 2001).

Marti Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the Fourteenth International Conference on Computational Linguistics*, Nantes, France.

Judith L. Klavans and Smaranda Muresan. 2000. DEFINDER: Rule-based methods for the extraction of medical terminology and their associated definitions from on-line text. In *Proceedings of the Annual Fall Symposium of the American Medical Informatics Association*.

Judith L. Klavans and Smaranda Muresan. 2001. Evaluation of the DEFINDER system for fully automatic glossary construction. In *Proceedings of AMIA Symposium 2001*.

Véronique Malaisé, Pierre Zweigenbaum, and Bruno Bachimont. 2004. Detecting semantic relations between terms in definitions. In S. Ananadiou and P. Zweigenbaum, editors, *COLING 2004 CompuTerm 2004: 3rd International Workshop on Computational Terminology*, pp. 55–62, Geneva, Switzerland. COLING.

Małgorzata Marciniak, Agnieszka Mykowiecka, Anna Kupść, and Jakub Piskorski. 2005. Intelligent content extraction from Polish medical texts. In L. Bolc et al., editors, *Intelligent Media Technology for Communicative Intelligence, Second International Workshop, IMTCI 2004, Warsaw, Poland, September 13-14, 2004, Revised Selected Papers*, volume 3490 of *Lecture Notes in Computer Science*, pp. 68–78. Springer-Verlag.

Svetoslav Marinov and Joakim Nivre. 2005. A data-driven parser for Bulgarian. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories*, pp. 89–100, Barcelona.

Spyridoula Miliaraki and Ion Androutsopoulos. 2004. Learning to identify single-snippet answers to definition questions. In *Proceedings of COLING 2004*, pp. 1360–1366, Geneva, Switzerland. COLING.

Jennifer Pearson. 1996. The expression of definitions in specialised texts: a corpus-based analysis. In M. Gellerstam et al., editors, *Proceedings of the Seventh Euralex International Congress*, pp. 817–824, Göteborg.

Jakub Piskorski et al. 2004. Information extraction for Polish using the SProUT platform. In M. A. Kłopotek et al., editors, *Intelligent Information Processing and Web Mining*, pp. 227–236. Springer-Verlag, Berlin.

Kiril Simov and Petya Osenova. 2005. BulQA: Bulgarian-Bulgarian Question Answering at CLEF 2005. In *CLEF*, pp. 517–526.

Kiril Simov et al. 2001. CLaRK — an XML-based system for corpora development. In P. Rayson et al., editors, *Proceedings of the Corpus Linguistics 2001 Conference*, pp. 558–560, Lancaster.

Angelika Storrer and Sandra Wellinghoff. 2006. Automated detection and annotation of term definitions in German text corpora. In *Proceedings of LREC 2006*.

Hristo Tanev. 2004. Socrates: A question answering prototype for Bulgarian. In *Recent Advances in Natural Language Processing III, Selected Papers from RANLP 2003*, pages 377–386. John Benjamins.

Richard Tobin, 2005. *Lxtransduce, a replacement for fsgmatch*. University of Edinburgh. `http://www.cogsci.ed.ac.uk/~richard/ltxml2/lxtransduce-manual.html`.

Stephan Walter and Manfred Pinkal. 2006. Automatic extraction of definitions from German court decisions. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pp. 20–28, Sydney, Australia. Association for Computational Linguistics.

Marcin Woliński. 2004. *Komputerowa weryfikacja gramatyki Świdzińskiego*. Ph. D. dissertation, ICS PAS, Warsaw.

Daniel Zeman. 2004. *Parsing with a Statistical Dependency Model*. Ph. D. dissertation, Charles University, Prague.

Unsupervised Methods of Topical Text Segmentation for Polish

Dominik Flejter
University of Economics
al. Niepodległości 10
Poznań, Poland
D.Flejter@
kie.ae.poznan.pl

Karol Wieloch
University of Economics
al. Niepodległości 10
Poznań, Poland
K.Wieloch@
kie.ae.poznan.pl

Witold Abramowicz
University of Economics
al. Niepodległości 10
Poznań, Poland
W.Abramowicz@
kie.ae.poznan.pl

Abstract

This paper describes a study on performance of existing unsupervised algorithms of text documents topical segmentation when applied to Polish plain text documents. For performance measurement five existing topical segmentation algorithms were selected, three different Polish test collections were created and seven approaches to text preprocessing were implemented. Based on quantitative results (P_k and WindowDiff metrics) use of specific algorithm was recommended and impact of pre-processing strategies was assessed. Thanks to use of standardized metrics and application of previously described methodology for test collection development, comparative results for Polish and English were also obtained.

1 Introduction

Rapid development of Internet-based information services is marked by a proliferation of information available on-line. Even if the Web shifts towards multimedia content and structured documents, contents of the Web sources remains predominantly textual and poorly structured; an important fraction of this flood of plain text documents consists in multitopical documents. This abundance of complex but plain text or just visually structured documents (as is in case of most HTML files) creates a strong need for intelligent text processing including robust and efficient information extraction and retrieval.

One way of increasing efficiency of typical text processing tasks consists in processing separately text segments instead of whole documents. While different text segmentation strategies can be applied including splitting text into segments of equal length, using moving windows of constant size or discourse units (Reynar, 1998), division into topical segment is intuitively more justified. This approach is applicable in several IR and NLP areas including document indexing, automatic summarization and question answering (Choi, 2002).

For Information Extraction domain, two main application of topical document segmentation are related to documents pre-processing and supporting some basic tasks widely used by IE tools. Topical segmentation applied to individual documents as well as documents streams (e.g. dialogues of radio broadcast transcripts) is an initial step for further IE processing (Manning, 1998); when combined with segments labelling, classification or clustering (Allan et al., 1998) it allows to pre-select ranges of text to be mined for mentions of events, entities and relations relevant to users needs and available IE resources. This limits significantly size of text to be processed by IE methods (Hearst and Plaunt, 1993) and thus influences significantly overall IE performance.

On the other hand, many basic tasks required by IE domain (both related to IE resources creation and document (pre-)processing) make use of sections rather than whole documents. In these tasks including language modelling (esp. gathering of co-occurrences statistics for trigger-based language models), anaphora resolution, word sense disambiguation and coreference detection, definition of proper context is crucial. To this extend entities

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 51–58,
Prague, June 2007. ©2007 Association for Computational Linguistics

discovered by topical segmentation are more reliable than document, paragraph or sentence contexts as they help avoid usage of unrelated parts of documents as well as minimize sparse data problems (Choi, 2002).

1.1 Problem Statement

In this study we adopted definition proposed in (Flejter, 2006) stating that "*linear topical segmentation of text consists in extracting coherent blocks of text such that: 1) any block focuses on exactly one topic 2) any pair of consecutive blocks have different main topics*." We also accepted that two segments should be defined as having different or same topics whenever they are judged so by people.

Depending on the used document collection and user needs, text segmentation objective is to find topical segments in individual multi-topical text documents or to discover boundaries between consecutive stories or news items (e.g. in radio broadcast transcriptions or text news streams).

Approaches to the problem of text segmentation can be divided according to a number of dimensions (Flejter, 2006) including cognitive approach (in optimistic approach text structure intended by author is cognitively accessible, in pessimistic approach it is not), hierarchical or linear segmentation (do we segment text into some levels of embedded segments?), completeness of segmentation (does the segmentation need to cover the whole text?), disjointness of segments (can two segments have any common text ranges?), fuzziness of segments boundaries (how fuzzy is the actual boundary location?), global or local view of topics (is there any global, document-independent list of topics?).

In this study we investigate linear, complete, disjoint segments with binary boundaries and local view of topics. Selected algorithms focus on text segmentation without considering labelling or clustering of discovered segments.

1.2 Our Contributions

The contributions of our research described in this paper are twofold. Firstly, we developed three Polish test collections for text segmentation task (see: Section 3.1). Secondly, we performed an extensive study of performance of most popular segmentation algorithms (see: Section 3.2) and pre-processing strategies (see: Section 3.3) when applied to Polish documents; a total of 42 scenarios including different algorithms, pre-processing strategies and test collections were evaluated (see: Section 4).

2 Approaches to Topical Segmentation

As in case of most NLP tasks, a number of different linguistic theories influenced topic segmentation resulting in a variety of approaches applied. This section gives a short presentation of major theories underlying topical segmentation and an overview of most popular segmentation algorithms with emphasis on those evaluated in our experiment.

2.1 Theoretical Foundations

Out of linguistic approaches cohesion theory of Halliday and Hassan had the strongest impact on topical text segmentation. It analyzes several mechanisms of documents internal cohesion including references, substitution, ellipsis, conjunction (logical relations) and lexical cohesion (reuse of the same words to address the same object or objects of the same class as well as use of terms which are more general or semantically related in systematic or nonsystematic way). Other relevant linguistic theories include Grosz and Sinder's discourse segmentation theory, Rhetorical Structure Theory and taxonomy of text structures proposed Skorochod'ko (Reynar, 1998).

Out of empirical statistical rules some authors make use of heuristics resulting form Heaps' law for new-words-based topical segment boundary detection. However, the most important theoretical foundations in quantitative methods are related to strong probabilistic frameworks including Hidden Markov Models (Mulbregt et al., 1998) and Maximal Entropy Theory (Beeferman et al., 1997).

2.2 Basic Methods

The most simple but also the most frequently used methods of topical text segmentation do not require training (thus they are domain-independent) nor make use of any complex linguistic resources or utilities. Apart from methods based on new vocabulary analysis, this category of algorithms applies widely the simplest form of lexical cohesion i.e. reiterations of the same word.

The first classical text segmentation algorithm of this type is TextTiling described in (Hearst and Plaunt, 1993). Its analysis unit consists of pseudo-sentences corresponding to series of consecutive words (typically 20 words). After the whole text is divided into pseudo-sentences, a window of 12 pseudo-sentences is slid over the text (with one pseudo-sentence step). At any position the window is decomposed into two six-pseudo-sentences blocks and their similarity is calculated by means of cosine measure. Measurements for all consecutive window positions (understood as positions of centre of the window) form lexical cohesion curve local minima of which correspond to segments boundaries. The original algorithm was further enhanced in several ways including use of words similarity measurement based on co-occurrences (Kaufmann, 1999).

Another group of basic algorithms makes use of technique of DotPlotting, originally proposed by Raynar in (Reynar, 1994). In this approach 2D chart is used for lexical cohesion analysis with both axes corresponding to positions (in words) in the text; on the chart points are drawn at coordinates (x, y) and (y, x) iff words at positions x and y are equal. In this settings coherent text segments correspond visually to squares with high density of points. DotPlotting image is than segmented using one of two strategies: minimization of points density at the boundaries (minimization of external incoherence) or maximization of density of segments (maximization of internal coherence) (Reynar, 1998). The original DotPlotting algorithm requires to explicitly provide expected number of segments as input.

Improved version of DotPlotting algorithm called C99 (Choi, 2000) uses DotPlotting chart for visualization of similarity measurements at consecutive point of the text (thus resulting in point with different levels of intensity) instead of words co-occurences. Afterwards, mask-based ranking technique is used for image enhancement. For actual segmentation, dynamic programming technique similar to DotPlotting maximization algorithm is used; an optional automatic termination strategy is also implemented thus allowing the algorithm to assess number of boundaries. In further work of the same and other authors several enhancements of C99 algorithm were proposed.

2.3 Methods Requiring External Resources

Still not requiring training and domain independent, some methods make use of linguistic resources more sophisticated than stop-list. Two classes of such solution described in existing work are solutions using lexical chains (Morris and Hirst, 1991; Min-Yen Kan, 1998) (which require to use some thesaurus) and based on spreading activation (Kozima, 1993) (which depend on weights-based semantic network constructed from thesaurus). In both cases the effort put in algorithm enactment is quite high; however in principle no additional resources need to be developed for new texts (even from different domains).

2.4 Methods Requiring Training

Last group of methods includes supervised methods with generally strong mathematical foundations. They perform very well; however they require training that possibly needs to be repeated when new domain needs to be addressed. The methods in this group use probabilistic frameworks including maximal entropy (Beeferman et al., 1997), Hidden Markov Models (Mulbregt et al., 1998) and Probabilistic Latent Semantic Analysis (Blei and Moreno, 2001).

3 Experimental Setup

Segmentation algorithms performance was evaluated for 42 scenarios corresponding to different algorithms, pre-processing strategies and test collections. For quantitative analysis and comparability with previous and future research results two standard segmentation metrics were applied in all scenarios.

3.1 Test Collections

For performance measurement three test collections corresponding to different types of segmentation tasks were developed: artificial documents collection (AC), stream collection (SC) and individual documents collection (DC). AC and SC were constructed based on 936 issues of EuroPAP (European information service of Polish Press Agency) plain-text e-mail newsletter (EuroPAP, 2005) collected from November 2001 to May 2005. Typical issue of EuroPAP newsletter contained about

25 complete news articles and a number of short (containing at most several sentences) news items. DC was constructed based on articles retrieved form Wikipedia corresponding to ten most populous Polish cities (Wikipedia, 2007) covering typically several topics (e.g. geography, culture, transportation).

For AC creation we followed precisely the method applied for English in (Choi, 2000). Each artificial document was created as a concatenation of random number (n) of first sentences from ten news articles randomly selected from a total of 24927 news items in EuroPAP corpus. Four subcollections were created depending on allowed range of n as listed in Table 1. Any two selected articles were assumed to cover two different topics; thus reference segmentation boundaries corresponded to points of concatenations.

AC	AC1	AC2	AC3	AC4
n	3-11	3-5	6-8	9-11
documents count	400	100	100	100

Table 1: Artificial collection subcollections

For SC creation newsletter messages from EuroPAP were used as text streams (936 messages). The reference segmentation was created using original article boundaries present in EuroPAP mail messages (almost 30000 segments were marked).

For individual documents collection development text content was extracted from Wikipedia documents, all headings were removed and all list items (LI tags) with no terminal punctuation sign were added a dot. Manual tagging by two authors of this paper was performed. The instructions were to put segment boundaries in the places of potential section titles. Obtained percent agreement of 0.988 and κ coefficient (Carletta, 1996) of 0.975 suggest high convergence of both annotations. Further, in places where the two annotators opinions differed (one marked boundary and the other did not), negotiation-based approach (Flejter, 2006) was applied in order to develop reference segmentation.

3.2 Selected algorithms

In our experiment we used Choi's publicly available implementation of several text segmentation algorithms not requiring training (with several adaptation concerning pre-processing stages). Specifically we used Choi's implementation of TextTiling algorithm (TT), C99 algorithm for both known ($C99_l$) and unknown ($C99$) number of boundaries as well as DotPlotting maximization (DP) and minimization (DP_{min}) algorithms.

	Sentences		Tokens	
	avg	std	avg	std
AC	6.8	1.7	122.6	33.4
SC	15.0	3.8	267.6	64.0
DC	28.5	9.9	300.0	110.2

Table 2: The average length of reference segments

Algorithms not requiring to provide number of segments as input (TT, $C99$) were evaluated on all test collections; performance of the remaining algorithms ($C99_l$, DP, DP_{min}) was measured only for AC.

3.3 Pre-processing variants

We decided to prepare seven variants of the test collections (see Table 3). The motivation for the first group (variants: P1, P2, P3, P4) was to be as close to Choi's methodology as possible. That's why we used simple pre-processing techniques like lemmatization and stop-lists. The remaining variants (P5, P6, P7) were chosen arbitrarily to check how additional morphological information will influence the performance of the main segmentation algorithms in case of Polish language.

The pre-processing stage included two steps. Initially, documents were split into sentences and word tokens (punctuation signs were removed) by means of tokenizer and sentence boundary recognizer of SProUT — a shallow text processing system tailored to processing Polish language (Piskorski et al., 2004). Afterwards the generated token stream was normalized; for this task SProUT's interface for a dictionary based Polish morphological analyzer — Morfeusz (Woliński, 2007) was used. This allowed us to use variety of morphological information (including STEM, POS, NUMBER, TENSE). The drawback of such an approach was that tokens not present in Morfeusz's dictionary were not stemmed (accounting for 12.8% of all tokens or 31.7% of unique tokens; note that Morfeusz input

Id	Variant	Description
P1	I	no changes (tokens remain inflected)
P2	L	lemmatized tokens
P3	LSL	L − words in the lemmatized stop-list
P4	ISI	I − without words in the inflected stop-list
P5	L-VT	L + verbs tagged with POS and TENSE
P6	L-VT-N-A	L-VT + nouns and adjectives tagged with POS
P7	L-VT-NN-AN	L-VT-N-A + nouns and adjectives tagged with NUMBER

Table 3: pre-processing variants

contained all the tokens including numbers, hyperlinks, dates, etc.). Another problematic issue was ambiguity of morphological analysis (1.4 interpretation on average); we addressed this issue by using the following order of preference: 1) verbs, 2) nouns, 3) adjectives, 4) other word classes.

The stop-lists (inflected and lemmatized versions contained 616 and 350 tokens respectively) were prepared manually by analysing frequency lists of previously used text corpus.

3.4 Evaluation metrics

A number of different measurement methods were applied to topical texts segmentation including recall-precision pair (Hearst and Plaunt, 1993; Passonneau and Litman, 1997), edit distance (Ponte and Croft, 1997), P_μ (Beeferman et al., 1997), P_k (Beeferman et al., 1999) and *WindowDiff* (Pevzner and Hearst, 2002).

P_k is simplified version of probabilistic measure P_μ based on assumption that any two consecutive boundaries are at distance of k sentences (k being parameter normally set to half of length of average segment in reference segmentation). After some simplifications P_k is defined by the following formula (Flejter, 2006):

$$P_k(r,h) = 1 - \frac{1}{n-k}\sum_{i=1}^{n-k}\left(|\delta_r(i,k) - \delta_h(i,k)|\right)$$

where $\delta_X(i,k)$ equals to one if ith and $(i+k)$th sentences are in the same segment of segmentation X, otherwise it is equal to zero; $X = r$ corresponds to reference segmentation and $X = h$ corresponds to hypothetical (algorithm-generated) segmentation.

In most publication instead of performance measurement using P_k, probabilistic error metric ($P = 1 - P_k(r,h)$) is applied. For easier comparison with previous results we calculated this measure for tested evaluation scenarios.

Based on a profound analysis of P_k and probabilistic metric drawbacks more recently WindowDiff error measure was proposed based on counting number of boundaries within window of size of k sentences sliding parallelly over both hypothetical and reference segmentations. WindowDiff can be calculated by the following formula:

$$W_k(r,h) = \frac{1}{N-k}\sum_{i=1}^{N-k}\left(|b_r(i,k) - b_h(i,k)| > 0\right)$$

with $b_X(i,k)$ corresponding to the number of boundaries between positions i and $i+k$ in segmentation X.

Both probabilistic error and WindowDiff measure the segmentation error; therefore, the lower is their value, the better is segmentation result.

4 Experimental Results

Calculated values of P and WindowDiff (WD) measures were used to compare performance of different algorithms, collections and pre-processing strategies. If not stated otherwise, results displayed in this Section correspond to P1 variant (no pre-processing) of test collections.

	$C99$		TT	
	P	WD	P	WD
AC	0.365	0.355	0.527	0.638
AC1	0.360	0.350	0.539	0.639
AC2	0.387	0.360	0.435	0.436
AC3	0.370	0.360	0.549	0.650
AC4	0.359	0.364	0.551	0.821
SC	0.381	0.390	0.562	0.932
DC	0.429	0.477	0.554	0.877

Table 4: Comparison of methods not requiring number of segments as input

Comparative results of both algorithms not requiring to provide expected number of segments as input (i.e. $C99$ and TT) are listed in Table 4. $C99$

performs much better than TextTiling on all test collections with extremely high WindowDiff value for TextTiling (especially for longer texts). Both algorithms perform better on artificial than on actual documents; for $C99$ drop in performance between stream and cities documents is also visible.

	$C99_l$		DP		DP_{min}	
	P	WD	P	WD	P	WD
AC	0.339	0.332	0.353	0.338	0.462	0.485
AC1	0.342	0.336	0.368	0.353	0.460	0.483
AC2	0.324	0.304	0.343	0.318	0.455	0.483
AC3	0.342	0.337	0.347	0.332	0.464	0.487
AC4	0.338	0.341	0.310	0.300	0.475	0.495

Table 5: Comparison of methods requiring number of segments as input

Probabilistic error and WindowDiff results for algorithms requiring expected number of segments as input ($C99_l$, DP, DP_{min}) are listed in Table 5. $C99_l$ performs slightly better than DotPlotting with maximization strategy (DP) and the performance of DotPlotting applying minimization strategy (DP_{min}) is visibly lower. Results do not differ significantly between AC subcollections suggesting that length of document has minor impact on performance.

As $C99$ was developed both in version requiring and not requiring to specify the expected number of segments, impact of this information on algorithm performance was analyzed. As expected $C99_l$ outperforms $C99$; in our experiments additional information on segments count lowered the error rates by 4–16% (see: Table 6).

As previous research on English text segmentation (Choi, 2000) was led for the same artificial collection creation methodology (see section 3.1), algorithm implementations and probabilistic error metric, comparison of algorithms performance for Polish and English was possible. The results of such comparison are gathered in Table 7; for English results were taken from Choi's paper and for Polish P3 variant of our artificial collection corresponding to Choi's pre-processing approach was used. Comparative analysis shows that all algorithms (except for TT which is highly inefficient for both Polish and English) perform significantly worse for Polish. Our hypothesis is that it can be attributed both to lower performance of pre-processing tools for Polish and usage of domain specific corpus as opposed to balanced Brown corpus used by Choi.

	$C99$		$C99_l$		$\Delta\%$	
	P	WD	P	WD	P	WD
AC	0.365	0.355	0.339	0.332	7%	6%
AC1	0.360	0.350	0.342	0.336	5%	4%
AC2	0.387	0.360	0.324	0.304	16%	16%
AC3	0.370	0.360	0.342	0.337	8%	6%
AC4	0.359	0.364	0.338	0.341	6%	6%

Table 6: Impact of segments count provided as input

		3-11	3-5	6-8	9-11
$C99(P3)$	PL	0.32	0.34	0.31	0.29
	EN	0.13	0.18	0.10	0.10
$C99_l(P3)$	PL	0.30	0.27	0.29	0.28
	EN	0.12	0.12	0.09	0.09
$DP(P3)$	PL	0.32	0.28	0.26	0.24
	EN	0.22	0.21	0.18	0.16
$DP_{min}(P3)$	PL	0.46	0.46	0.46	0.47
	EN	n/a	0.34	0.37	0.37
$TT(P3)$	PL	0.50	0.39	0.49	0.54
	EN	0.54	0.45	0.52	0.53

Table 7: Polish versus English results

	$AC + C99_l$		$SC + C99$		$DC + C99$	
	P	WD	P	WD	P	WD
P1	0.365	0.355	0.381	0.390	0.429	0.477
P2	0.322	0.315	0.356	**0.367**	0.433	0.477
P3	**0.319**	**0.311**	**0.354**	0.368	0.452	0.497
P4	0.342	0.334	0.381	0.391	0.425	0.473
P5	0.362	0.318	0.356	0.368	0.435	0.480
P6	0.416	0.403	0.413	0.419	**0.406**	**0.430**
P7	0.416	0.403	0.413	0.419	**0.406**	**0.430**
$\Delta\%$	13%	12%	7%	6%	5%	10%

Table 8: Impact of different pre-processing variants

Final part of our analysis of quantitative results focused on impact of different pre-processing strategies. For each of three test collections we analyzed the impact of pre-processing strategies in case of best performing algorithm.

As the results from Table 8 show, for artificial and stream collections the most promising strategy is to use P3 (LSL) variant of pre-processing which decreased the error metric by up to 13% as compared with P1 (no pre-processing) variant. Interestingly, the same approach applied to individual documents collection actually increased values of error metrics; in this case significant decrease of error rates was possible by use of the most complex P6 and P7 strategies. Reasons for this difference may include: a) disproportion in number of unrecognized tokens (22.8% for DC vs. 12.75% for AC/SC), b) different structure of DC reference segments (higher number of shorter sentences, see: Table 2), c) standard deviation of segment length in DC much higher than in AC/DC (35% of average length in case of DC vs. 25% in case of both AC and SC). We leave analysis of this factors' impact for future work.

Our analysis also shows that adding NUMBER tag for nouns and adjectives (P7) has no impact on algorithms performance as compared with P6.

5 Conclusions and Future Work

In this paper we analyzed performance of several topical text segmentation algorithms for Polish with several pre-processing strategies on three different test collections. Our research demonstrate that similarly to English $C99$ (and its variant with expected segments count input $C99_l$) is the best performing segmentation algorithm and we recommend that it be applied to text segmentation for Polish. Based on our research we also suggest that lemmatization and stop-list words removal (P3 variant) be used for further improvement of performance. However, our research revealed that the performance of almost all algorithms (including $C99$) is significantly worse for Polish than for English and remains unsatisfactory.

Therefore our further research direction will be to focus on improvements at the pre-processing stages of text segmentation (including enhancements in text division into sentences, lemmatization of proper names, and filtering of unrecognized tokens with low document-based frequency) as well as on analysis of performance of more recent algorithms both requiring and not requiring linguistic resources. We would like also to evaluate text segmentation impact on performance of coreference resolution algorithm we are currently developing.

References

James Allan, Jaime Carbonell, George Doddington, Jonathan Yamron, and Yiming Yang. 1998. Topic detection and tracking pilot study. final report.

Doug Beeferman, Adam Berger, and John Lafferty. 1997. Text segmentation using exponential models. In Claire Cardie and Ralph Weischedel, editors, *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 35–46. Association for Computational Linguistics, Somerset, New Jersey.

Doug Beeferman, Adam Berger, and John Lafferty. 1999. Statistical models for text segmentation. *Machine Learning*, 34(1-3):177–210.

David M. Blei and Pedro J. Moreno. 2001. Topic segmentation with an aspect hidden Markov model. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348, New York, NY, USA. ACM Press.

Jean Carletta. 1996. Assessing agreement on classification task: the kappa statistic. *Computational Linguistic*, 22:250–254.

Freddy Y. Y. Choi. 2000. Advances in domain independent linear text segmentation. In *Proceedings of the first conference on North American chapter of the Association for Computational Linguistics*, pages 26–33, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Freddy Y. Y. Choi. 2002. *Content-based Text Navigation*. Ph.D. thesis, Department of Computer Science, University of Manchester.

EuroPAP. 2005. Serwis o Unii Europejskiej, http://euro.pap.com.pl/ (2001-2005).

Dominik Flejter. 2006. Automatic topical segmentation of documents in text collections (in polish). Master's thesis, Poznan University of Economics, June.

Marti A. Hearst and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 59–68, New York, NY, USA. ACM Press.

Stefan Kaufmann. 1999. Cohesion and collocation: Using context vectors in text segmentation. In *Proceedings of the 37th annual meeting of the Association for*

Computational Linguistics on Computational Linguistics, pages 591–595, Morristown, NJ, USA. Association for Computational Linguistics.

Hideki Kozima. 1993. Text segmentation based on similarity between words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, pages 286–288, Morristown, NJ, USA. Association for Computational Linguistics.

C. Manning. 1998. Rethinking text segmentation models: An information extraction case study. Technical report, University of Sydney.

Kathleen R. McKeown Min-Yen Kan, Judith L. Klavans. 1998. Linear segmentation and segment significance. In *Proceedings of 6th International Workshop of Very Large Corpora (WVLC-6)*, pages 197–205.

Jane Morris and Graeme Hirst. 1991. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48.

P. Mulbregt, I. van, L. Gillick, S. Lowe, and J. Yamron. 1998. Text segmentation and topic tracking on broadcast news via a hidden markov model approach. In *Proceedings of the ICSLP'98, volume 6*, pages 2519–2522.

Rebecca J. Passonneau and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Comput. Linguist.*, 23(1):103–139.

Lev Pevzner and Marti A. Hearst. 2002. A critique and improvement of an evaluation metric for text segmentation. *Comput. Linguist.*, 28(1):19–36.

Jakub Piskorski, Peter Homola, Malgorzata Marciniak, Agnieszka Mykowiecka, Adam Przepiórkowski, and Marcin Wolinski. 2004. Information extraction for Polish using the SProUT platform. In *Intelligent Information Processing and Web Mining, Proceedings of the International Intelligent Information Systems: IIPWM'04 Conference*, Advances in Soft Computing, pages 227–236. Springer.

Jay M. Ponte and W. Bruce Croft. 1997. Text segmentation by topic. In *ECDL '97: Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 113–125, London, UK. Springer-Verlag.

Jeffrey C. Reynar. 1994. An automatic method of finding topic boundaries. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 331–333, Morristown, NJ, USA. Association for Computational Linguistics.

Jeffrey C. Reynar. 1998. *Topic segmentation: algorithms and applications. A dissertation in Computer and Information Science*. Ph.D. thesis, University of Pennsylvania.

Wikipedia. 2007. Miasta w Polsce wedlug liczby ludnosci, Accessed on 20.03.2007.

Marcin Woliński. 2007. Analizator morfologiczny Morfeusz SIAT. On-line. Accessed on: 30.01.2007.

Multi-word Term Extraction for Bulgarian

Svetla Koeva
Department of Computational Linguistics – IBL
Bulgarian Academy of Sciences
52 Shipchenski prohod Blv. Sofia 1113, Bulgaria
svetla@ibl.bas.bg

Abstract

The goal of this paper is to compile a method for multi-word term extraction, taking into account both the linguistic properties of Bulgarian terms and their statistical rates. The method relies on the extraction of term candidates matching given syntactic patterns followed by statistical (by means of Log-likelihood ratio) and linguistically (by means of inflectional clustering) based filtering aimed at improving the coverage and the precision of multi-word term extraction.

1 Introduction

The goal of this paper is to compile a method for multi-word term extraction, taking into account both the linguistic properties of Bulgarian terms and their statistical rates. Term extraction exploits well-established techniques that seem difficult to improve significantly. As in many other areas of computational linguistics, term extraction has been approached generally with three different strategies – linguistic techniques, statistical techniques and a combination of both (Bourigault *et al.*, 2001; Jacquemin & Bourigault, 2000). The linguistically based techniques exploit the morpho-syntactic structure of terms that usually differ from one language to another (for example in Bulgarian and in English the most frequent syntactic structure representing terms is the noun phrase, but the two languages significantly differ in their constituent structure and agreement properties). The automatic extraction of term morpho-syntactic patterns, being in most cases language-dependent, requires specific language processing – Part-of-speech (POS) tagging, lemmatization, syntactic parsing, etc. The statistical techniques, on the other hand, rely on the different statistical features of terms compared to other words in the text and are usually based on the detection of words and expressions with a frequency value higher than a given limit. Some of the statistical approaches focus on the association measures between the components of the multi-word terms. Hybrid approaches, combining linguistic and statistical techniques, are also applied, mainly in two manners: statistical proceeding is used to filter the term candidates obtained through linguistic techniques, and, vice versa, some linguistic filters are exploited after statistical processing, in order to extract the statistically significant word combinations that match some given syntactic patterns.

The method for automatic multi-word term extraction, presented in this paper, also relies both on linguistic knowledge and on statistical processing. The research aims are to:

- Apply syntactic patterns of Bulgarian terms directed to multi-word term extraction;
- Use well-known statistical methods (association measures) to eliminate some of the irrelevant multi-word terms;
- Further limit the number of invalid terms by clustering term candidates around their lemmas;
- Test the performance of such a method over the manually annotated corpus.

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 59–66,
Prague, June 2007. ©2007 Association for Computational Linguistics

Most of the current methods for automatic term extraction are developed for English, and thus they are not appropriate for direct adaptation to Bulgarian, due to the morpho-syntactic differences between the two languages. Bulgarian is a language with a rich inflectional system. That is to say, a noun lemma can appear in six forms if it is masculine and in four forms if it is feminine or neuter. Besides, noun phrase structure and agreement properties in Bulgarian differ in some aspects from other languages such as English, Therefore, a language-specific approach is needed if we want to utilise the morpho-syntactical information for term extraction. To the best of our knowledge there is no report of an extensive work directed towards Bulgarian term extraction.

The structure of our paper outlines the three steps involved in our approach. In the following section we present a short linguistic analysis of Bulgarian terms. In the third section, we describe the identification of the candidate terms. The fourth section explains how we applied a list of terms to the filters. We then evaluate our results on a corpus that was set up by manual annotation. Finally, we discuss some peculiarities of the presented study and propose future works to be done.

2 Linguistic analysis of Bulgarian terms

2.1. Compilation of a term annotated corpus

We share the views that larger corpora not only give statistically more reliable counts, but also reveal phenomena that are completely lacking in smaller samples. The Acquis Communautaire (AC)[1] – the European Union legislation, which consists of approximately eight thousand documents containing approximately 40 million words (to be more specific, its Bulgarian subpart) – is targeted as the most appropriate resource for our research: because of its size, and because of the number of languages included in it. (The proposed method can be further transformed and/or evaluated to deal with the rest of the languages represented in the parallel corpus.)

The AC contains documents from several domains, which are divided into chapters: Agriculture, Fisheries, Transport Policy, Taxation, Economic and Monetary Union, Statistics, Social Policy and Employment, Energy, Industrial Policy, Education and Training, Telecommunication and Information Technologies, Culture and Audiovisual Policy, etc. This annotated subpart of the Bulgarian AC is developed as a test corpus and contains 10,521 words from randomly selected text samples representing the domains of Agriculture (AGR), Energy (ENR) and Education and Training (EDC).

Some criteria for the manual annotation of Bulgarian terms were defined, the notion of term among others. As with most linguistic concepts, a term is defined in various ways. For example, as "a word or expression that has a precise meaning in some uses or is peculiar to a science, art, profession" (Webster, 2002), or as "a word or expression used for some particular thing"[2], or generally as words or phrases that denote specific concepts in a given subject domain. For the purposes of this investigation we defined a term as

An open class word or expression that is peculiar to a specific domain of human activities and occurs with a determinate (in some limits) frequency in that domain.

The annotation of terms in the Bulgarian AC subpart is also based on both the maximum and minimum length term selection. That is, in the case of a multi-word term which constituents are also terms, the longest term (as well as all shorter terms) is selected. It should be pointed out, however, that the term annotated corpus is still small enough to be representative of the word frequency and is a sample of translated texts that might manifest different tendencies for a term's distribution from those in the original texts.

2.2. Single-word terms vs. multi-word terms

The general impression is that the most of the papers dealing with automatic term extraction (especially the statistically based ones) are focused on multi-word terms. This can be explained by the fact that for English a bigger percentage of multi-word terms comparing to single-word terms is reported. To show the tendency for the correlation between single-word and multi-word terms in Bulgarian texts, the manually annotated subpart of the Bulgarian AC has been studied. We found out (Table 1.) that the proportion of single-word terms

[1] There has been some experience of exploiting the AC as a multilingual corpus (Steinberger at al., 2006).

[2] http://wordnet.princeton.edu

varies from about 2.5% to 3% depending on the subject domain.

The results show that the use of single-word terms in Bulgarian technical documents is also not very frequent and the tendency is that multi-word terms are preferred to single-word ones. Following these observations, first we will concentrate on the extraction of the Bulgarian multi-word terms.

Domain	AGR	ENR	EDC	Total
#Words	4423	3002	3096	10521
#Terms (T)	344	297	254	895
#Multi-word T	266	165	171	602
#Single-word T	111	89	93	293
% Terms	7,77	9,89	8,2	8,5
% Single-word T	2,5	2,96	3	2,78

Table 1. Distribution of single-word terms

2.3 Syntactic structures of Bulgarian terms

The starting point for the linguistically motivated part of the automatic term extraction is to describe the syntactic structure of Bulgarian terms. There are several Bulgarian terminological dictionaries published and some terminological databases available on the internet – all recourses are taken into consideration in the analysis without providing exact calculations. The collection of Bulgarian terms, obtained by the annotated subpart of the Bulgarian AC, is used as a source for the determination of the most frequent syntactic structures of Bulgarian terms.

It is claimed that NPs constitute about 80-99 % of whole terms in an English text, with the varying percentage depending on the text types (Arppe, 1995). The same statement is roughly true for Bulgarian; although there are some adjectives and verbs that can be regarded as terms in a certain domain (only three verbs and one adjective are detected in the annotated corpus). In this study we have concentrated on the NPs' term extraction, which comprises the focus of interest in several studies (Jacquemin, 2001; Justeson & Katz, 1995; Voutanen, 1993).

In order to obtain the statistics, the annotated part of Bulgarian AC is pre-processing. This allows the consequences of the categories constituting Bulgarian terms to be extracted and their frequency to be calculated. As a result, 16 different sequences of categories are obtained, among them 5 with a rate higher than 11 %. In the next examples the most frequent syntactic patterns of the Bulgarian multi-word terms are listed following their frequency rate:

- AN → *riboloven sezon* (fishing season), *iglolistno darvo* (conifer), *zemedelski ceni* (firm prices), *termalna energiya* (thermal energy), *klimatichna instalaciya* (air-conditioning);
- NpN → *obogatyavane na gorivo* (fuel enrichment), *podobryavane na pochvata* (soil improvement), *prava na deteto* (children's rights), *svoboda na pechata* (freedom of the press);
- NpAN → o*pazvane na okolnata sreda* (environmental protection), *nomenklatura na zemedelskite produkti* (agricultural product nomenclature), *izpolzvane na slanchevata energiya* (solar energy end-use applications), *sredstva za masova informaciya* (media);
- AAN → *semeyno zemedelsko stopanstwo* (family farming), *evropeyska parichna sistema* (European Monetary System), *inteligentna transportna sistema* (intelligent transport system), *magniten informacionen nositel* (magnetic medium);
- ANpN → *elektronen transfer na fondove* (electronic funds transfer), *opticheskо razpoznavane na simvoli* (Optical Character Recognition), *pravna uredba na telekomunikaciite* (regulation of telecommunications), *izbiratelno razprostranenie na informaciya* (selective dissemination of information).

Among the five types, the AN structure was the most frequent one, although the exact percentage still remains to be calculated over the bigger corpus.

The main differences observed concerning these five Bulgarian structures and their English equivalents are the regular agreement between the adjectival modifier and the head noun in Bulgarian and the prepositional phrase in Bulgarian instead the noun modifier in English. The adjective-noun agreement in Bulgarian noun phrases is partially exploited in the presented piece of work, but it might be extensively considered in further improvements of the method.

In the case of NpN, NpAN and ANpN structures, we found out that most of the terms corresponding to these patterns are built up with the Bulgarian preposition *na* (of). This may be explained by the fact that these PPs usually correspond to the English NPs with a noun modifier denoting more specific concepts. The possible strings of categories that might constitute the Bulgarian terms are exploited due to the fact that Bulgarian terms usually do not allow other constituents among their parts.

2.4 Term variations

Some authors have pointed out the discrepancy between term representation in dictionaries, and the term forms used in real texts (Daille, 2003). It is well known that the same concept can be formulated in different ways and the automatic term extraction should be able to recognize and link those different linguistic forms or expressions. Different kinds of term variants are distinguished in the literature: orthographic variants (capitalization), inflectional variants (word forms), morpho-syntactic variants (derivation), syntactic variants (word order differences) and semantic variants (synonyms).

In this study only the orthographic and inflectional variants are taken into consideration. It should be pointed out that compared to lemmas the multi-word terms have their own inflective rules. The POS of the head word determines the clustering of the term into grammatical classes, such as noun, adjective, and so on, which define the possible slots in the paradigm.

The significant grammatical categories inherent to the lemma of the head word (such as gender for nouns), the number and POS of the remaining constituents and the options for inserting some words (such as particles) in the multi-word term structure all show the grouping of multi-word terms' grammatical subclasses and define which slots of the paradigm are realized in the language. And finally, the formation of word forms of each component of a multi-word term and the type of agreement dependencies between components show the classification of multi-word terms into grammatical types that describe the real word paradigm belonging to a particular term (Koeva, 2005).

For instance, the Bulgarian term *klimatichna instalaciya* (air-conditioning) is a noun phrase; the members of the paradigm are determined by the head feminine noun. The inflection type is determined by the inflectional alternations of each member (the adjective and the noun):

klimatichna instalaciya	– singular, indefinite
klimatichnata instalaciya	– singular, definite
klimatichni instalaciii	– plural, indefinite
klimatichnite instalaciii	– plural, definite

There are agreement dependencies between adjective and head noun and no other words' intervention or word order changes are allowed.

3 Automatic term extraction

3.1 Pre-processing of the Bulgarian AC

It is common practice to extract candidate terms using a part-of-speech (POS) tagger and an automaton (a program extracting word sequences corresponding to predefined POS patterns). The part-of-speech tagging is the process of automatically identifying the words in a text as corresponding to a particular part of speech. The part-of-speech tagger used in this study is developed utilizing a large manually annotated corpus consisting of 197,000 tokens (150,000 words) randomly extracted from the Bulgarian Brawn corpus (1,000,000 words) (Koeva *et al.,* 2006). The tagger has been developed as a modified version of the Brill tagger (Brill, 1994). The Brill tagger was trained for Bulgarian using a part of the tagged corpus. We applied a rule-based approach leading to 98.3% precision. A sophisticated tokenizer that recognizes sentence boundaries and categorizes tokens as words, abbreviations, punctuation, numerical expressions, hours, dates and URLs has been built as a part of the tagger. For each word in the text the initial (most probable) part of speech among the ambiguity set is assigned from a large inflectional dictionary (Koeva, 1998).

The words that are not recognized by the dictionary are handled by the guesser analyzing the suffixes of the unrecognized words and assigning the initial part of speech among the ambiguity set. The part-of-speech ambiguity ratio calculated over the annotated corpus is 1.51 tags per word, which means that on average every second word is ambiguous. For solving the ambiguity, 144 contextual rules are implemented, utilizing the part of speech and dictionary information on the context, Some additional techniques for the optimizations are implemented – the application of dictionaries of abbreviations, proper nouns, grammatically unambiguous words, etc. After POS tagging the text remains unchanged and the additional information is added in an xml format.

Lemmatization is the process of automatic determining the lemma for a given word. Since the lemmatization involves fixing the part of speech of a word, it requires the running of a tagger. Lemmatization is closely related to stemming. The difference is that a stemmer operates on a single word without explicit knowledge of its identity as a part of speech, its lemma or its inflectional properties. For Bulgarian a large inflectional dictionary is used both for lemmatization and stemming.

The tag sets differ both in how the words are divided into categories, and in how their categories are defined. For the purposes of this investigation the grammatical information characterizing the forms is also assigned to nouns and adjectives, because the adjective-noun agreement is exploited.

3.2 Extraction of term candidates

Following the frequency analysis of the constituent structure of the Bulgarian multi-word terms, the targeted syntactic patterns will be recognized by the following regular expression:

[(A+N(pA*N)?)(NpA*N)]

The strings of categories bellow will be matched; those with more than two adjectives are either rare, or not observed in the language:

AN, AAN, NpN, NpAN, ANpN, ANpAN, NpAAN, ANpAAN, AANpAAN, …

The regular expression does not match the single Ns as well as the NPs with low frequently – only the five syntactic patterns with the highest frequency rate are targeted for the term extraction. Moreover, the agreement features of the Bulgarian NP structures are exploited considering the unification of grammatical features between the preceding adjective and the immediate following adjective or noun. Based on patterns' matching, the term candidates corresponding to the above regular expressions are extracted:

- AN → *osnovno obrazovanie* (basic education),
- AAN → *novi obrazovatelni metodi* (new educational methods), *evropeyska audiovizualna zona* (European audiovisual area),
- NpN → *ezik za programirane* (programming language),
- NpAN → *planirane na uchebnata godina* (planning of the school year), *elekronna obrabotka na danni* (electronic data processing), *potrebitel na ingormacionna tehnologiya* (information technology user), etc.

On the other hand, the following phrases (which are annotated as terms) are not recognized:

- NpVpN → *aparat za vazproizwodstvo na zvuk* (sound reproduction equipment),
- AcAN → *poshtenski i telekomunikacionni uslugi* (postal and telecommunications services),
- NpNpNN → *sistema za upravlenie na baza danni* (database management system), etc.

A deficiency of the approach based on the syntactic patterns is also the fact that any NP that matches the patterns will be selected as a term candidate, as is shown in the following examples:

- AN → *novi metodi* (new methods), *ogranicheno dvizhenie* (limited circulation),
- NpN → *analiz na informaciya* (information analysis), *broy na uchenicite* (number of pupils), etc.

Some of the noun phrases are wrongly extracted, although in this case this is concerned with a compositional building of structures that cannot be considered as that of multi-word terms. Some term candidates with a preposition cannot be treated even as phrases, because their parts belong to different sentence constituents. The identification of the sub-phrases that are themselves also terms should also be taken into account. In the following example, *sistema za upravlenie na baza ot danni* (database management system), the phrases *sistema za uprawlenie* (management system), *upravlenie na baza ot danni* (database management) and *baza ot danni* (database) are also terms.

Domain	AGR	ENG	EDC	Total
#Words	4,423	3,002	3,096	10,521
#Term candidates	901	778	712	2,391

Table 2. Number of term candidates

The number of extracted term candidates depends on the structure of the sentences that occur in the selected domains. Table 2 shows the extracted term candidates from a Bulgarian AC subpart representing texts from the Agriculture, Energy and Education domains.

4 Filtering of term candidates

As a filtering mechanism we adopted the calculating of the associativity between words, which is often used to identify word collocations, and the term clustering according to the inflexional paradigms.

4.1 Statistical filtering

The frequency-based techniques applied to term filtering assign a numerical value to sets of words to rank term candidates and exclude those term candidates below a certain threshold. The statement that the more frequently a lexical unit appears in a given document the more likely it is that this unit has a terminological function can be applied to certain genres of texts. Alone, frequency is not a robust metric for assessing the terminological property of a candidate.

In our case, we want to measure the cohesion of a multi-word candidate term by verifying if its words occur together as a coincidence or not. Association measures are often used to rate the correlation of word pairs (Daille, 1995; Daille *et al.*, 1998).

	B	!B	
A	N_{ii}	N_{ij}	N_{1p}
!A	N_{ji}	N_{jj}	N_{2p}
	N_{p1}	N_{p2}	N_{pp}

Table 3. The contingency table

These measures can be derived from the contingency table (Table 3.) of the word pair (A,B) containing the observed frequencies of (A,B), as follows:

N_{ii} = the joint frequency of word A and word B;

N_{ij} = the frequency word A occurs and word B does not;

N_{ji} = the frequency word B occurs and word A does not;

N_{jj} = the frequency word A and word B do not occur;

N_{pp} = the total number of ngrams;

N_{p1}, N_{p2}, N_{1p}, N_{2p} are the marginal counts.

The lexical association measures are formulas that relate the observed frequencies to the expected frequency (Mij = (Np1 * N1p) / Npp) under the assumption that A and B are independent. For the current work, the Log-likelihood coefficient has been employed (Dunning, 1993), as it is reported to perform well among other scoring methods (Daille, 1995).

$$\text{Log-likelihood} = 2 * \sum (N_{ij} * \log(N_{ij} / M_{ij}))$$

This calculation over the text serves as an important technique in identifying term candidates. The larger the value of Log-likelihood is, the stronger is the association between the two pairs of the string; consequently the string is the most probable candidate. Statistic filtering is applied only to those term candidates extracted by the linguistic component. For the calculation, the Ngram Statistics Package (NSP), programs that aids in analyzing ngrams, is executed (Banerjee & Pedersen, 2003). The NSP takes text files (in our case Cyrillic letters are transliterated into Latin) as input and generates a list of bigrams along with their frequencies as outputs. Over the list of bigrams obtained, the Log-likelihood is run to compute a ratio for each ngram. The bigrams are targeted because some of the term candidates initially extracted are long ones containing sub-phrases that are likely to function as term candidates. In order to avoid potential term candidates being included in other longer phrases, the term candidates are split and the constituting bigrams are generated.

As a result of statistical filtering, the initially selected term candidates are assigned different values according to their word association. The Log-likelihood coefficient computed for each bigram is used to decide whether or not there is enough evidence to reject or accept a bigram - there is a clear opposition between small and big values. Below the first five ranked candidates are listed.

1. *evropeyskata obshtnost* (European community)
2. *atomna energiya* (nuclear energy)
3. *detska gradina* (kindergarten)
4. *Darzhaven vestnik* (government newspaper)
5. *obrazovatelna sistema* (educational system)

4.2 Linguistic filtering

The linguistic filtering aims at linking the different variations of the same basic term. The list of the automatically extracted terms was reviewed by means of lemmatization in order to refine it and to increase the score of some terms. Until this stage the different word forms of a term were calculated separately. Bulgarian is a highly inflected language – the forms of the head noun can vary from one to seven depending of the gender, number and references to a person. The sequences of lemmas belonging to the term candidates are processed and the frequency values are recalculated according to the grouping of terms in one inflectional cluster with respect to the common canonical form. Through this technique morphologically-related occurrences, such as *iglolistno darvo* (a conifer), *iglolistnoto darvo* (the conifer), *iglolistni darveta* (conifers) and *iglolistnite darveta* (the conifers) are treated as one term.

5 Evaluation

The presented method of identifying Bulgarian multi-word terms was applied on the manually annotated corpus. First the texts were pre-processed by means of POS tagging and lemmatization, then the target syntactic patterns were extracted, and the rates of the related bigrams were calculated by means of Log-likelihood association, and finally additional reordering of term candidates was performed by means of inflectional clustering. As a result, 430 (from 539) correctly extracted multi-word terms are obtained – the precision of 79.96% is registered.

6 Conclusions and future work

We have presented a method aimed at extracting Bulgarian multi-word terms, which relies on the extraction of syntactic patterns from text and on the statistical and linguistically based filtering aimed at improving the coverage and the precision of multi-word collocation extraction. We have applied Log-likelihood ratio statistical filtering to the extracted multi-word terms. All extracted term candidates are grammatically correct, due to the syntactically based pattern matching. Further developments of the method include:

- Statistical determination of single-word terms;
- Coverage of long-distance occurrence and rare syntactic structures of multi-word terms;
- Analyzing the embedded terms.
- Using 'stop lists' of open and closed class words that are hardly to be found in the multi-word terms.

Some other experiments will be made using other well-known techniques of association measure. For the evaluation purposes the test corpus will be extended. A bigger homogeneous corpus would undoubtedly result in an increase in terms with more representative frequencies, and, therefore, in an improvement in statistical estimation of terms. The results can be exploited in the multilingual term extraction, due to the fact that the AC represents the biggest multilingual parallel corpus.

References

A. Aprre 1995. Term Extraction from Unrestricted Text: *10th Nordic Conference of Computational Linguistics (NoDaLiDa)*, Helsinki.

S. Banerjee and T. Pedersen 2003. The Design, Implementation, and Use of the Ngram Statistics Package, *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City.

D. Bourigault, C. Jacquemin, and M.-C. L'Homme 2001. *Recent Advances in Computational Terminology*, volume 2 of Natural Language Processing, John Benjamins.

E. Brill 1994. Some Advances In Rule-Based Part of Speech Tagging *AAAI,* Seattle, Washington

B. Daille 1995. *Combined approach for terminology extraction: lexical statistics and linguistic filtering*. Technical paper. UCREL, Lancaster University.

B. Daille 2003. Conceptual structuring through term variations, *Proceedings of the ACL Workshop on Multiword Expressions: Analysis, Acquisition and Treatment.*

B. Daille, E. Gaussier, and J.-M. Lange 1998. An Evaluation of Statistical Scores for Word Association, in J. Ginzburg, Z. Khasidashvili, C. Vogel, J.-J. Levy, and E. Vallduvi (eds), *The Tbilisi Symposium on Logic, Language and Computation: Selected Papers*, CSLI Publications, p. 177-188.

T. Dunning 1993. Accurate methods for thestatistics of surprise and coincidence,. *Computational Linguistics*, 19(1):61–74.

C. Jacquemin 2001. *Spotting and Discovering Terms through Natural Language Processing*. MIT Press.

C. Jacquemin and D. Bouricault 2000. Chapter 19 Term Extraction and Automatic Indexing, *Handbook of Computational Linguistics* (R. Mitkov (ed.*))*, Oxford University Press, Oxford.

J. S. Justeson and S. M. Katz 1995. Technical Terminology: Some Linguistic Properties and an Algorithm for Identification in Text, *Natural Language Engineering*. 1(1):9-27.

S. Koeva 1998. Bulgarian Grammatical dictionary. Organization of the language data, *Bulgarian language*, vol. 6: 49-58.

S. Koeva 2005. Inflection Morphology of Bulgarian Multiword Expressions, *Computer Applications in Slavic Studies – Proceedings of Azbuki@net, International Conference and Workshop*, Sofia, 201-216.

S. Koeva, S. Leseva, I. Stoyanova, E. Tarpomanova, and M. Todorova 2006. Bulgarian Tagged Corpora, *Proceedings of the Fifth International Conference Formal Approaches to South Slavic and Balkan Languages,* Sofia, 78-86.

R. Steinberger, B. Pouliquen, A. Widiger, C. Ignat, T. Erjavec, D. Tufiş, and D. Varga 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages, *Proceedings of the 5th International Conference on Language Resources and Evaluation* (LREC'2006), Genoa.

A. Voutilainen. 1993. NPtool. A detector of English noun phrases, *Proceedings of the Workshop on Very Large Corpora*, Columbus, Ohio.

The Best of Two Worlds: Cooperation of Statistical and Rule-Based Taggers for Czech

Drahomíra "johanka" Spoustová
Jan Hajič
Jan Votrubec
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics,
Charles University Prague, Czech Republic
{johanka,hajic,votrubec}@
ufal.mff.cuni.cz

Pavel Krbec
IBM Czech Republic,
Voice Technologies and Systems,
Prague, Czech Republic,
pavel_krbec@cz.ibm.com

Pavel Květoň
Institute of the Czech Language,
Academy of Sciences of the Czech Republic
Pavel.Kveton@seznam.cz

Abstract

Several hybrid disambiguation methods are described which combine the strength of hand-written disambiguation rules and statistical taggers. Three different statistical (HMM, Maximum-Entropy and Averaged Perceptron) taggers are used in a tagging experiment using Prague Dependency Treebank. The results of the hybrid systems are better than any other method tried for Czech tagging so far.

1 Introduction

Inflective languages pose a specific problem in tagging due to two phenomena: highly inflective nature (causing sparse data problem in any statistically based system), and free word order (causing fixed-context systems, such as n-gram HMMs, to be even less adequate than for English).

The average tagset contains about 1,000 – 2,000 distinct tags; the size of the set of possible and plausible tags can reach several thousands. There have been attempts at solving this problem for some of the highly inflective European languages, such as (Daelemans, 1996), (Erjavec, 1999) for Slovenian and (Hajič, 2000) for five Central and Eastern European languages.

Several taggers already exist for Czech, e.g. (Hajič et al., 2001b), (Smith, 2005), (Hajič et al., 2006) and (Votrubec, 2006). The last one reaches the best accuracy for Czech so far (95.12 %). Hence no system has reached – in the absolute terms – a performance comparable to English tagging (such as (Ratnaparkhi, 1996)), which stands above 97 %.

We are using the Prague Dependency Treebank (Hajič et al., 2006) (PDT) with about 1.8 million hand annotated tokens of Czech for training and testing. The tagging experiments in this paper all use the Czech morphological (pre)processor, which includes a guesser for "unknown" tokens and which is available from the PDT website (PDT Guide, 2006) to disambiguate only among those tags which are morphologically plausible.

The meaning of the Czech tags (each tag has 15 positions) we are using is explained in Table 1. The detailed linguistic description of the individual positions can be found in the documentation to the PDT (Hajič et al., 2006).

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 67–74,
Prague, June 2007. ©2007 Association for Computational Linguistics

	Name	Description
1	POS	Part of Speech
2	SUBPOS	Detailed POS
3	GENDER	Gender
4	NUMBER	Number
5	CASE	Case
6	POSSGENDER	Possessor's Gender
7	POSSNUMBER	Possessor's Number
8	PERSON	Person
9	TENSE	Tense
10	GRADE	Degree of comparison
11	NEGATION	Negation
12	VOICE	Voice
13	RESERVE1	Unused
14	RESERVE2	Unused
15	VAR	Variant

Table 1: Czech Morphology and the Positional Tags

2 Components of the hybrid system

2.1 The HMM tagger

The HMM tagger is based on the well known formula of HMM tagging:

$$\hat{T} = \arg\max_T P(T)P(W \mid T) \tag{1}$$

where

$$\begin{aligned} P(W|T) &\approx \textstyle\prod_{i=1}^{n} P(w_i \mid t_i, t_{i-1}) \\ P(T) &\approx \textstyle\prod_{i=1}^{n} P(t_i \mid t_{i-1}, t_{i-2}). \end{aligned} \tag{2}$$

The trigram probability $P(W \mid T)$ in formula 2 replaces (Hajič et al., 2001b) the common (and less accurate) bigram approach. We will use this tagger as a baseline system for further improvements.

Initially, we change the formula 1 by introducing a scaling mechanism[1]: $\hat{T} = \arg\max_T(\lambda_T * logP(T) + logP(W \mid T))$.

We tag the word sequence from right to left, i.e. we change the trigram probability $P(W \mid T)$ from formula 2 to $P(w_i \mid t_i, t_{i+1})$.

Both the output probability $P(w_i \mid t_i, t_{i+1})$ and the transition probability $P(T)$ suffer a lot due to the data sparseness problem. We introduce a component $P(ending_i \mid t_i, t_{i+1})$, where $ending$ consists of the last three characters of w_i. Also, we introduce another component $P(t_i^* \mid t_{i+1}^*, t_{i+2}^*)$ based on a reduced tagset T^* that contains positions POS, GENDER, NUMBER and CASE only (chosen on linguistic grounds).

We upgrade all trigrams to fourgrams; the smoothing mechanism for fourgrams is history-based bucketing (Krbec, 2005).

The final fine-tuned HMM tagger thus uses all the enhancements and every component contains its scaling factor which has been computed using held-out data. The total error rate reduction is 13.98 % relative on development data, measured against the baseline HMM tagger.

2.2 Morče

The Morče[2] tagger assumes some of the HMM properties at runtime, namely those that allow the Viterbi algorithm to be used to find the best tag sequence for a given text. However, the transition weights are not probabilities. They are estimated by an Averaged Perceptron described in (Collins, 2002). Averaged Perceptron works with features which describe the current tag and its context.

Features can be derived from any information we already have about the text. Every feature can be true or false in a given context, so we can regard current true features as a description of the current tag context.

For every feature, the Averaged Perceptron stores its weight coefficient, which is typically an integer number. The whole task of Averaged Perceptron is to sum all the coefficients of true features in a given context. The result is passed to the Viterbi algorithm as a transition weight for a given tag. Mathematically, we can rewrite it as:

$$w(C,T) = \sum_{i=1}^{n} \alpha_i . \phi_i(C,T) \tag{3}$$

where $w(C,T)$ is the transition weight for tag T in context C, n is number of features, α_i is the weight coefficient of i^{th} feature and $\phi(C,T)_i$ is evaluation of i^{th} feature for context C and tag T.

Weight coefficients (α) are estimated on training data, cf. (Votrubec, 2006). The training algorithm is very simple, therefore it can be quickly retrained and it gives a possibility to test many different sets of features (Votrubec, 2005). As a result, Morče gives the best accuracy from the standalone taggers.

[1] The optimum value of the scaling parameter λ_T can be tuned using held-out data.

[2] The name Morče stands for "MORfologie ČEštiny" ("Czech morphology").

2.3 The Feature-Based Tagger

The Feature-based tagger, taken also from the PDT (Hajič et al., 2006) distribution used in our experiments uses a general log-linear model in its basic formulation:

$$p_{AC}(y \mid x) = \frac{\exp(\sum_{i=1}^{n} \lambda_i f_i(y,x))}{Z(x)} \qquad (4)$$

where $f_i(y,x)$ is a binary-valued feature of the event value being predicted and its context, λ_i is a weight of the feature f_i, and the $Z(x)$ is the natural normalization factor.

The weights λ_i are approximated by Maximum Likelihood (using the feature counts relative to all feature contexts found), reducing the model essentially to Naive Bayes. The approximation is necessary due to the millions of the possible features which make the usual entropy maximization infeasible. The model makes heavy use of single-category Ambiguity Classes (AC)[3], which (being independent on the tagger's intermediate decisions) can be included in both left and right contexts of the features.

2.4 The rule-based component

The approach to tagging (understood as a standalone task) using hand-written disambiguation rules has been proposed and implemented for the first time in the form of Constraint-Based Grammars (Karlsson, 1995). On a larger scale, this aproach was applied to English, (Karlsson, 1995) and (Samuelsson, 1997), and French (Chanod, 1995). Also (Bick, 2000) uses manually written disambiguation rules for tagging Brazilian Portuguese, (Karlsson, 1985) and (Koskenniemi, 1990) for Finish and (Oflazer, 1997) reports the same for Turkish.

2.4.1 Overview

In the hybrid tagging system presented in this paper, the rule-based component is used to further reduce the ambiguity (the number of tags) of tokens in an input sentence, as output by the morphological processor (see Sect. 1). The core of the component is a hand-written *grammar* (set of rules).

Each rule represents a portion of knowledge of the language system (in particular, of Czech). The knowledge encoded in each rule is formally defined in two parts: a sequence of tokens that is searched for in an input sentence and the tags that can be deleted if the sequence of tokens is found.

The overall strategy of this "negative" grammar is to keep the highest recall possible (i.e. 100 %) and gradually improve precision. In other words, whenever a rule deletes a tag, it is (almost) 100% safe that the deleted tag is "incorrect" in the sentence, i.e. the tag cannot be present in any correct tagging of the sentence.

Such an (virtually) "error-free" grammar can partially disambiguate any input and prevent the subsequent taggers (stochastic, in our case) to choose tags that are "safely incorrect".

2.4.2 The rules

Formally, each rule consists of the description of the *context* (sequence of tokens with some special property), and the *action* to be performed given the context (which tags are to be discarded). The length of context is not limited by any constant; however, for practical purposes, the context cannot cross over sentence boundaries.

For example: in Czech, two finite verbs cannot appear within one clause. This fact can be used to define the following disambiguation rule:

- context: unambiguous finite verb, followed/preceded by a sequence of tokens containing neither a comma nor a coordinating conjunction, at either side of a word x ambiguous between a finite verb and another reading;
- action: delete the finite verb reading(s) at the word x.

It is obvious that no rule can contain knowledge of the whole language system. In particular, each rule is focused on at most a few special phenomena of the language. But whenever a rule deletes a tag from a sentence, the information about the sentence structure "increases". This can help other rules to be applied and to delete more and more tags.

For example, let's have an input sentence with two finite verbs within one clause, both of them ambiguous with some other (non-finite-verbal) tags. In this situation, the sample rule above cannot be applied.

[3] If a token can be a N(oun), V(erb) or A(djective), its (major POS) Ambiguity Class is the value "ANV".

On the other hand, if some other rule exists in the grammar that can delete non-finite-verbal tags from one of the tokens, then the way for application of the sample rule is opened.

The rules operate in a loop in which (theoretically) all rules are applied again whenever a rule deletes a tag in the partially disambiguated sentence. Since deletion is a monotonic operation, the algorithm is guaranteed to terminate; effective implementation has also been found in (Květoň, 2006).

2.4.3 Grammar used in tests

The grammar is being developed since 2000 as a standalone module that performs Czech morphological disambiguation. There are two ways of rule development:

- the rules developed by syntactic introspection: such rules are subsequently verified on the corpus material, then implemented and the implemented rules are tested on a testing corpus;
- the rules are derived from the corpus by introspection and subsequently implemented.

In particular, the rules are not based on examination of errors of stochastic taggers.

The set of rules is (manually) divided into two (disjoint) reliability classes — *safe* rules (100% reliable rules) and *heuristics* (highly reliable rules, but obscure exceptions can be found). The safe rules reflect general syntactic regularities of Czech; for instance, no word form in the nominative case can follow an unambiguous preposition. The less reliable heuristic rules can be exemplified by those accounting for some special intricate relations of grammatical agreement in Czech.

The grammar consists of 1727 safe rules and 504 heuristic rules. The system has been used in two ways:

- *safe rules only*: in this mode, safe rules are executed in the loop until some tags are being deleted. The system terminates as soon as no rule can delete any tag.
- *all rules*: safe rules are executed first (see *safe rules only* mode). Then heuristic rules start to operate in the loop (similarly to the safe rules). Any time a heuristic rule deletes a tag, the *safe rules only* mode is entered as a subprocedure. When safe rules' execution terminates, the loop of heuristic rules continues. The disambiguation is finished when no heuristic rule can delete any tag.

The rules are written in the *fast LanGR* formalism (Květoň, 2006) which is a subset of more general LanGR formalism (Květoň, 2005). The LanGR formalism has been developed specially for writing and implementing disambiguation rules.

3 Methods of combination

3.1 Serial combination

The simplest way of combining a hand-written disambiguation grammar with a stochastic tagger is to let the grammar reduce the ambiguity of the tagger's input. Formally, an input text is processed as follows:

1. morphological analysis (every input token gets all tags that are plausible without looking at context);
2. rule-based component (partially disambiguates the input, i.e. deletes some tags);
3. the stochastic tagger (gets partially disambiguated text on its input).

This algorithm was already used in (Hajič et al., 2001b), only components were changed — the ruled-based component was significantly improved and two different sets of rules were tried, as well as three different statistical taggers. The best result was (not surprisingly) achieved with set of safe rules followed by the Morče tagger.

An identical approach was used in (Tapanainen, 1994) for English.

3.2 Serial combination with SUBPOS pre-processing

Manual inspection of the output of the application of the hand-written rules on the development data (as used in the serial combination described in the previous section) discovered that certain types of deadlocked ("cross-dependent") rules prevent successful disambiguation.

Cross-dependence means that a rule A can not apply because of some remaining ambiguity, which could be resolved by a rule B, but the operation of B is still dependent on the application of A. In particular, ambiguity in the Part-of-Speech category is very problematic. For example, only a few safe rules can apply to a three-word sentence where all three words are ambiguous between finite verbs and something else.

If the Part-of-Speech ambiguity of the input is already resolved, precision of the rule-based component and also of the final result after applying any of the statistical taggers improves. Full Part-of-Speech information is represented by the first two categories of the Czech morphology tagset — POS and SUBPOS, which deals with different types of pronouns, adverbs etc. As POS is uniquely determined by SUBPOS (Hajič et al., 2006), it is sufficient to resolve the SUBPOS ambiguity only.

All three taggers achieve more than 99% accuracy in SUBPOS disambiguation. For SUBPOS disambiguation, we use the taggers in usual way (i.e. they determine the whole tag) and then we put back all tags having the same SUBPOS as the tag chosen by the tagger.

Thus, the method with SUBPOS pre-processing operates in four steps:

1. morphological analysis;
2. SUBPOS disambiguation (any tagger);
3. rule-based component;
4. final disambiguation (the same tagger[4]).

The best results were again achieved with the tagger Morče and set of safe rules.

3.3 Combining more taggers in parallel

This method is quite different from previous ones, because it essentially needs more than one tagger. It consists of the following steps:

1. morphological analysis;
2. running N taggers independently;
3. merging the results from the previous step — each token ends up with between 1 and N tags, a union of the taggers' outputs;
4. (optional: the rule-based component;)
5. final disambiguation (single tagger).

The best results were achieved with two taggers in Step 1 (Feature-based and Morče), set of all rules in Step 3 and the HMM tagger in Step 4.

This method is based on an assumption that different stochastic taggers make complementary mistakes, so that the recall of the "union" of taggers is almost 100 %. Several existing language models are based on this assumption — (Brill, 1998) for tagging English, (Borin, 2000) for tagging German and (Vidová-Hladká, 2000) for tagging inflective languages. All these models perform some kind of "voting" — for every token, one tagger is selected as the most appropriate to supply the correct tag. The model presented in this paper, however, entrusts the selection of the correct tag to another tagger that already operates on the partially disambiguated input.

4 Results

All the methods presented in this paper have been trained and tested on the PDT version 2.0[5]. Taggers were trained on PDT 2.0 training data set (1,539,241 tokens), the results were achieved on PDT 2.0 evaluation-test data set (219,765 tokens), except Table 6, where PDT 2.0 development-test data set (201,651 tokens) was used. The morphological analysis processor and all the taggers were used in versions from April 2006 (Hajič et al., 2006), the rule-based component is from September 2006.

For evaluation, we use both precision and recall (and the corresponding F-measure) and accuracy, since we also want to evaluate the partial disambiguation achieved by the hand-written rules alone. Let t denote the number of tokens in the test data, let c denote the number of tags assigned to all tokens by a disambiguation process and let h denote

[4] This limitation is obviously not necessary, but we treat this combination primarily as a one-tagger method. Results of employing two different taggers are only slightly better, but still much worse than results of other methods presented later below.

[5] The results cannot be simply (number-to-number) compared to previous results on Czech tagging, because different training and testing data (PDT 2.0 instead of PDT 1.0) are used since 2006.

the number of tokens where the manually assigned tag is present in the output of the process.

- In case of the morphological analysis processor and the standalone rule-based component, the output can contain more than one tag for every token. Then *precision* (p), *recall* (r) and *F-measure* (f) characteristics are defined as follows:

$$p = h/c \qquad r = h/t \qquad f = 2pr/(p + r).$$

- The output of the stochastic taggers contains always exactly one tag for every token — then $p = r = f = h/t$ holds and this ratio is denoted as *accuracy*.

Table 2 shows the performance of the morphological analysis processor and the standalone rule-based component. Table 3 shows the performance of the standalone taggers. The improvement of the combination methods is presented in Table 4.

Table 5 shows the relative error rate reduction. The best method presented by this paper (parallel combination of taggers with all rules) reaches the relative error rate decrease of 11.48 % in comparison with the tagger Morče (which achieves the best results for Czech so far).

Table 6 shows error rate (100 % − *accuracy*) of various methods[6] on particular positions of the tags (13 and 14 are omitted). The most problematic position is CASE (5), whose error rate was significantly reduced.

5 Conclusion

We have presented several variations of a novel method for combining statistical and hand-written rule-based tagging. In all cases, the rule-based component brings an improvement — the smaller the involvement of the statistical component(s) is, the bigger. The smallest gain can be observed in the case of the parallel combination of taggers (which by itself brings an expected improvement). The best variation improved the accuracy of the best-performing standalone statistical tagger by over 11 % (in terms of relative error rate reduction), and the inclusion of the rule-component itself improved the best statistical-only combination by over 3.5 % relative.

This might actually lead to pessimism regarding the rule-based component. Most other inflective languages however have much smaller datasets available than Czech has today; in those cases, we expect that the contribution of the rule-based component (which does not depend on the training data size, obviously) will be much more substantial.

The *LanGR* formalism, now well-developed, could be used for relatively fast development for other languages. We are, of course, unable to give exact figures of what will take less effort — whether to annotate more data or to develop the rule-based component for a particular language. Our feeling is that the jury is actually still out on this issue, despite some people saying that annotation is always cheaper: annotation for morphologically complex (e.g., inflective) languages is *not* cheap, and rule-based development efforts have not been previously using (unannotated) corpora so extensively (which is what *LanGR* supports for "testing" the developed rules, leading to more reliable rules and more effective development cycle).

On the other hand, the rule-based component has also two obvious and well-known disadvantages: it is language dependent, and the application of the rules is slower than even the baseline HMM tagger despite the "fast" version of the *LanGR* implementation we are using[7].

In any case, our experiments produced a software suite which gives the all-time best results in Czech tagging, and we have offered to apply it to re-tag the existing 200 mil. word Czech National Corpus. It should significantly improve the user experience (for searching the corpus) and allow for more precise experiments with parsing and other NLP applications that use that corpus.

[6] *F-b* stands for feature-based taggeer, *Par* for parallel combination without rules and *Par+Rul* for parallel combination with rules.

[7] In the tests presented in this paper, the speed of the operation of each stochastic tagger (and the parallel combination without rules) is several hundreds of tokens processed per second (running on a 2.2GHz Opteron processor). The operation of the standalone rule-based component, however, is cca 10 times slower — about 40 tokens per second. The parallel combination with all rules processes about 60 tokens per second — the rules operate faster here because their input in parallel combination is already partially disambiguated.

Method	p	r	f
Morphology	25.72 %	99.39 %	40.87 %
Safe rules	57.90 %	98.83 %	73.02 %
All rules	66.35 %	98.03 %	79.14 %

Table 2: Evaluation of rules alone

Tagger	accuracy
Feature-based	94.04 %
HMM	94.82 %
Morče	95.12 %

Table 3: Evaluation of the taggers alone

Combination method	accuracy
Serial (safe rules+Morče)	95.34 %
SUBPOS serial (safe rules+Morče)	95.44 %
Parallel without rules	95.52 %
Parallel with all rules	95.68 %

Table 4: Evaluation of the combinations

Method	Morče	Parallel without rules
Parallel without rules	8.20 %	—
Parallel with all rules	11.48 %	3.57 %

Table 5: Relative error rate reduction

	F-b	HMM	Morče	Par	Par+Rul
1	0.61	0.70	0.66	0.57	0.57
2	0.69	0.78	0.75	0.64	0.64
3	1.82	1.49	1.66	1.39	1.37
4	1.56	1.30	1.38	1.18	1.15
5	4.03	3.53	3.08	2.85	2.62
6	0.02	0.03	0.03	0.02	0.02
7	0.01	0.01	0.01	0.01	0.01
8	0.06	0.07	0.08	0.06	0.05
9	0.05	0.08	0.07	0.05	0.04
10	0.29	0.28	0.30	0.26	0.27
11	0.29	0.31	0.33	0.28	0.28
12	0.05	0.08	0.06	0.05	0.04
15	0.31	0.31	0.31	0.28	0.29

Table 6: Error rate [%] on particular positions of tags

Acknowledgements

The research described here was supported by the projects *MSM0021620838* and *LC536* of *Ministry of Eduation, Youth and Sports* of the Czech Republic, *GA405/06/0589* of the *Grant Agency* of the Czech Republic and *1ET100610409 Diagnostic and Evaluation Tools for Linguistic Software* of the *Information Society Programme* of the *National Research Programme* of the Czech Republic.

References

Eckhard Bick. 2000. The parsing system "Palavras" — automatic grammatical analysis of Portuguese in a constraint grammar framework. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation, TELRI.* Athens

Lars Borin. 2000. Something borrowed, something blue: Rule-based combination of POS taggers. In: *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, Vol. 1, pp. 21–26. Athens

Eric Brill and Jun Wu. 1998. Classifier combination for improved lexical disambiguation. In: *Proceedings of the 17th international conference on Computational linguistics*, Vol. 1, pp. 191–195. Montreal, Quebec

Jean-Pierre Chanod and Pasi Tapanainen. 1995. Tagging French — comparing a statistical and a constraint-based method. In: *Proceedings of EACL-95*, pp. 149–157. Dublin

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In: *Proceedings of EMNLP'02*, July 2002, pp. 1–8. Philadelphia

W. Daelemans and Jakub Zavrel and Peter Berck and Steven Gillis. 1996. MBT: A memory-based part of speech tagger-generator. In: *Proceedings of the 4th WVLC*, pp. 14–27. Copenhagen

Tomaz Erjavec and Saso Dzeroski and Jakub Zavrel. 1999. Morphosyntactic Tagging of Slovene: Evaluating PoS Taggers and Tagsets. *Technical Report*, Dept. for Intelligent Systems, Jozef Stefan Institute. Ljubljana

Jan Hajič and Barbora Hladká. 1997. Tagging of inflective languages: a comparison. In: *Proceedings of ANLP '97*, pp. 136–143. Washington, DC.

Jan Hajič 2000. Morphological tagging: Data vs. dictionaries. In: *Proceedings of the 6th ANLP / 1st NAACL'00*, pp. 94–101. Seattle, WA

Jan Hajič, Pavel Krbec, Pavel Květoň, Karel Oliva and Vladimír Petkevič. 2001. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In: *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. CNRS – Institut de Recherche en Informatique de Toulouse and Université des Sciences Sociales, pp. 260–267. Toulouse

Jan Hajič, Eva Hajičová, Jarmila Panevová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka and Marie Mikulová. 2006. Prague Dependency Treebank v2.0. CDROM. Linguistic Data Consortium, Cat. LDC2006T01. Philadelphia. ISBN 1-58563-370-4. Documentation also at `http://ufal.ms.mff.cuni.cz/pdt2.0`.

Fred Karlsson. 1985. Parsing Finnish in terms of a process grammar. In: Fred Karlsson (ed.): *Computational Morphosyntax: Report on Research 1981-84*, University of Helsinki, Department of General Linguistics Publications No. 13, pp. 137–176.

Fred Karlsson and Atro Voutilainen and Juha Heikkilä and Arto Anttila (eds.). 1995. Constraint Grammar: a language-independent system for parsing unrestricted text. *Natural Language Processing*. Vol. 4, Mouton de Gruyter, Berlin and New York.

Kimmo Koskenniemi. 1990. Finite-State Parsing and Disambiguation. In: *Proceedings of Coling-90*, University of Helsinki, 1990, pp. 229–232. Helsinki

Pavel Krbec. 2005. *Language Modelling for Speech Recognition of Czech.* PhD Thesis, MFF, Charles University Prague.

Pavel Květoň. 2005. *Rule-based Morphological Disambiguation.* PhD Thesis, MFF, Charles University Prague.

Pavel Květoň. 2006. Rule-based morphological disambiguation: On computational complexity of the LanGR formalism. In: *The Prague Bulletin of Mathematical Linguistics*, Vol. 85, pp. 57–72. Prague

Kemal Oflazer and Gökhan Tür. 1997. Morphological disambiguation by voting constraints. In: *Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics*, pp. 222–229. Madrid

Karel Oliva, Milena Hnátková, Vladimír Petkevič and Pavel Květoň. 2000. The Linguistic Basis of a Rule-Based Tagger of Czech. In: Sojka P., Kopeček I., Pala K. (eds.): *Proceedings of the Conference "Text, Speech and Dialogue 2000", Lecture Notes in Artificial Intelligence*, Vol. 1902. Springer-Verlag, pp. 3–8. Berlin-Heidelberg

PDT Guide. `http://ufal.ms.mff.cuni.cz/pdt2.0`

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In: *Proceedings of the 1st EMNLP*, May 1996, pp. 133–142. Philadelphia

Christer Samuelsson and Atro Voluntainen. 1997. Comparing a linguistic and a stochastic tagger. In: *Proceedings of ACL/EACL Joint Converence*, pp. 246–252. Madrid

Noah A. Smith and David A. Smith and Roy W. Tromble. 2005. Context-Based Morphological Disambiguation with Random Fields. In: *Proceedings of HLT/EMNLP*, pp. 475–482. Vancouver

Drahomíra "johanka" Spoustová. in prep. *Kombinované statisticko-pravidlové metody značkování češtiny. (Combining Statistical and Rule-Based Approaches to Morphological Tagging of Czech Texts).* PhD Thesis, MFF UK, in prep.

Pasi Tapanainen and Atro Voutilainen. 1994. Tagging accurately: don't guess if you know. In: *Proceedings of the 4th conference on Applied Natural Language Processing*, pp. 47–52. Stuttgart

Barbora Vidová-Hladká. 2000. *Czech Language Tagging.* PhD thesis, ÚFAL MFF UK. Prague

Jan Votrubec. 2005. *Volba vhodných rysů pro morfologické značkování češtiny. (Feature Selection for Morphological Tagging of Czech.)* Master thesis, MFF, Charles University, Prague.

Jan Votrubec. 2006. Morphological Tagging Based on Averaged Perceptron. In: *WDS'06 Proceedings of Contributed Papers*, MFF UK, pp. 191–195. Prague

Derivational Relations in Czech WordNet

Karel Pala
Faculty of Informatics
Masaryk University Brno
Czech Republic
pala@fi.muni.cz

Dana Hlaváčková
Faculty of Informatics
Masaryk University Brno
Czech Republic
ydana@aurora.fi.muni.cz

Abstract

In the paper we describe enriching Czech WordNet with the derivational relations that in highly inflectional languages like Czech form typical derivational nests (or subnets). Derivational relations are mostly of semantic nature and their regularity in Czech allows us to add them to the WordNet almost automatically. For this purpose we have used the derivational version of morphological analyzer Ajka that is able to handle the basic and most productive derivational relations in Czech. Using a special derivational interface developed in our NLP Lab we have explored the semantic nature of the selected noun derivational suffixes and established a set of the semantically labeled derivational relations – presently 14. We have added them to the Czech WordNet and in this way enriched it with approx. 30 000 new Czech synsets. A similar enrichment for Princeton WordNet has been reported in its recently released version 3.0, we will comment on the partial similarities and differences.

1 Introduction

WordNets as such represent huge semantic networks in which the basic units – synsets – are linked with the ‚main‘ semantic relations like synonymy, near_synony my, antonymy, hypero/hyponymy, meronymy and others. In the EuroWordNet project (cf. Vossen, 2003) Internal Language Relations (ILR) have been introduced such as Role_Agent, Agent_Involved or Role_Patient, Patient_Involved etc., as well as the relation Derivative capturing derivational relations between synsets. The semantic nature of the derivational relations, however, was not systematically analyzed and labeled in EuroWordNet project.

If we try to label the derivational relations semantically and include them in WordNet as a result we get two level network where on the higher level we have the ‚main‘ semantic relations between synsets such as synonymy, near_synonymy, antonymy, hypero/hyponymy, meronymy and others and on the lower level there are relations like the derivational ones that hold rather between literals than between synsets.

In the highly inflectional languages the derivational relations represent a system of semantic relations that definitely reflects cognitive structures that may be related to a language ontology. Such ontology undoubtedly exists but according to our knowledge it has not been written down yet. However, for language users derivational affixes (morphemes) function as formal means by which they express semantic relations necessary for using language as a vehicle of communication. In our view, the derivational relations should be considered as having semantic nature though a question may be asked what kind of semantics we are dealing with (see Sect. 3). It has to be remarked that grammatical categories such as gender or number display a clear semantic nature.

2 Derivational Morphology in Czech

In Czech words are regularly inflected (declined, conjugated) as they express different grammatical categories (gender, number, case, person, tense, aspect etc.) using affixes. This is what is called

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 75–81,
Prague, June 2007.

formal morphology in Czech grammars and its description mostly deals with the system of the inflectional paradigms. Then there is a *derivational morphology* which deals with deriving words from other words, e.g. nouns from verbs, adjectives from nouns or verbs etc. using affixes again. The derivations are closely related to the inflectional paradigms in a specific way: we can speak about derivational paradigms as well (cf. Pala, Sedláček, Veber, 2003).

For Czech inflectional morphology there are automatic tools – morphological analyzers exploiting the formal description of the inflection paradigms – we work with the analyzer called Ajka (cf. Sedláček, Smrž, 2003) and developed in our NLP Lab. Its list of stems contains approx. 400 000 items, up to 1600 inflectional paradigms and it is able to generate approx. 6 mil. Czech word forms.

We are using it for lemmatization and tagging, as a module for syntactic analyzer, etc. We have also developed a derivational version of Ajka (D-Ajka) that is able to work with the main regular derivational relations in Czech – it can generate new word forms derived from the stems. Together with D-Ajka an MWE preprocessing module with the database containing approx. 100 000 collocations is exploited as well.

2.1 Derivational relations in Czech

The derivational relations (D-relations) in Czech cover a large part of the word stock (up to 70 %). Thus we are interested in describing derivational processes (see examples) by which new words are formed from the corresponding word bases (roots, stems). In Czech grammars (Mluvnice češtiny, 1986) we can find at least the following main types (presently 14) of the derivational processes:

1. mutation: noun -> noun derivation, e.g. *ryba -ryb-ník* (*fish -> pond*), semantic relation expresses location – between an object and its typical location,
2. transposition (existing between different POS): noun -> adjective derivation, e.g. *den -> den-ní* (*day ->daily*), semantically the relation expresses property,

3. agentive relation (existing between different POS): verb -> noun e.g. *učit -> uči-tel (teach -> teacher*), semantically the relation exists between action and its agent,

4. patient relation: verb -> noun, e.g. *trestat -> trestanec* (*punish ->convict*), semantically it expresses a relation between an action and the object (person) impacted by it,

5. instrument (means) relation: verb -> noun, e.g. *držet -> držák* (*hold ->holder*), semantically it expresses a tool (means) used when performing an action,

6. action relation (existing between different POS): verb -> noun, e.g. *učit -> uče-n-í* (*teach -> teaching*), usually the derived nouns are charaterized as deverbatives, semantically both members of the relation denote action (process),

7. property-va relation (existing between different POS): verb -> adjective, e.g. *vypracovat -> vypracova-ný* (*work out -> worked out*), usually the derived adjectives are labelled as de-adjectives, semantically it is a relation between action and its property,

8. property-aad relation (existing between different POS): adjective -> adverb, e.g. *rychlý -> rychl-e* (*quick -> quickly*), semantically we can speak about property,

9. property-an (existing between different POS): adjective -> noun, e.g. *rychlý -> rychl-ost* (*fast -> speed*), semantically the relation expresses property in both cases,

10. gender change relation: noun -> noun, e.g. *inženýr -> inženýr-ka* (*engineer -> she engineer*), semantically the only difference is in sex of the persons denoted by these nouns,

11. diminutive relation: noun -> noun -> noun, e.g. *dům -> dom-ek -> dom-eček* (*house -> small house -> very little house* or *a house to which a speaker has an emotional attitude*), in Czech the diminutive relation can be binary or ternary,

12. augmentative relation: noun -> noun, e.g. *bába -> bab-izna* (*beldame -> hag*), semantically it expresses different emotional attitudes to a person,

13. prefixation: verb -> verb, e.g *myslet -> vy-myslet* (*think -> invent*), semantically prefixes in Czech denote a number of

different relations such as distributive, location, time, measure and some others. We will not be dealing with this topic here, it calls for a separate examination (project),

14. possessive relation (existing between different POS): noun -> adjective *otec* -> *otcův* (*father* -> *father's*), semantically it is a relation between an object (person) and its possession.

We should mention two more relations that are sometimes regarded inflectional but in our view they belong here as well: gerund relation - verb -> adjective: (*bojovat* ->*bojující, fight* -> *fighting*) and passive relation – verb -> adjective (passive participle): (*učit* -> *učen*, *teach* -> *taught*).

These 14 (+2) relations have been taken as a starting point for including derivational relations in Czech Wordnet. The main condition for their including is whether they can be generated by the derivational version of the analyzer Ajka. In this way we have been able to obtain automatically a precise specification what literals are linked together. It was also necessary to introduce the labels for the individual relations in a more systematic way. As a result we have obtained the following list of 10 derivational relations with their semantic labels that are given in the brackets and hold between the indicated POS:

1. deriv-na: noun -> adjective (property)
2. deriv-ger: verb -> adjective (property)
3. deriv-dvrb: verb -> noun (activity as a noun)
4. deriv-pos: noun -> adjective (possessive relation)
5. deriv-pas: verb -> adjective (passive relation)
6. deriv-aad: adjective -> adverb (property of property)
7. deriv-an: adjective -> noun (property)
8. deriv-g: noun -> noun (gender relation)
9. deriv-ag: verb -> noun (agentive relation)
10. deriv-dem: noun -> noun (diminutive relation)

The location and patient relation will be included in CzWn when the D-Ajka will be able to handle them (in the near future).

2.2 Derivational nests – subnets

If we have a look at the data, i.e. at the list of Czech stems and affixes and try to see how the just described relations work we obtain the typical derivational clusters – we will prefer to call them derivational nests (subnets). To illustrate their regularity we adduce an example of such nest for the Czech roots – *prác/prac-* (*work*). The main relations holding between these roots and the corresponding suffixes are:

roots: -prác-/-prac-e-

deriv-act - *prac-ova-t* (*to work*)

deriv-loc1- *prac-ov-iště* (*workplace*)

deriv-loc2 - *prac-ov-na* (*study*)

deriv-ag1- *prac-ov-ník* (*worker*),

deriv-g - *prac-ovn-ice* (*she-worker*),

deriv-ag2 - *prac-ant* (*plodder*)

deriv-ger - *prac-uj-ící* (*working - person*)

deriv-pro - *prac-ov-ní* (*professional, working*)

deriv-pro - *prac-ov-i-t-ý* (*diligent, hardworking*)

deriv-pro - *prac-ov-i-t-ost* (*diligence*)

The proposed labels are not final yet – the number of the productive derivational relations that have to be examined in Czech is larger, certainly up to 15. Number of the derivational suffixes in Czech is higher – more than 80.

At the moment the derivational Ajka is not able to generate the full nests automatically but we continue processing the remaining Czech derivational suffixes for this purpose.

2.3 Processing derivational suffixes

So far we have not said much about the affixes, i.e. prefixes, stem-forming infixes and suffixes used in derivations. In this analysis we pay attention mainly to the suffixes, prefixes are related mostly to verbs and in this sense they represent a separate and rather complicated derivational system. Infixes or intersegments are basically covered by the list of stems – instead writing rules for changes in stems we just use more variants of one stem. But the root analysis is possible and if we want to describe the derivational processes in Czech as completely as possible we have to return to them.

As starting data we have used a list of noun stems taken from the stem dictionary of the D-Ajka analyzer – their number is approx. 126 000. The derivations have been analyzed by means of the web interface developed just for this purpose. Noun derivations are performed in the three basic steps:

1. a set of words is defined by means of the (prefix), suffix and morphological tag;
2. defining a derivational rule – typically a substitution of morphemes (suffixes) at the end of the word;
3. manual modification of the results – usually correcting or deleting cases that cannot be regarded as properly derived forms though they may follow the given rule.

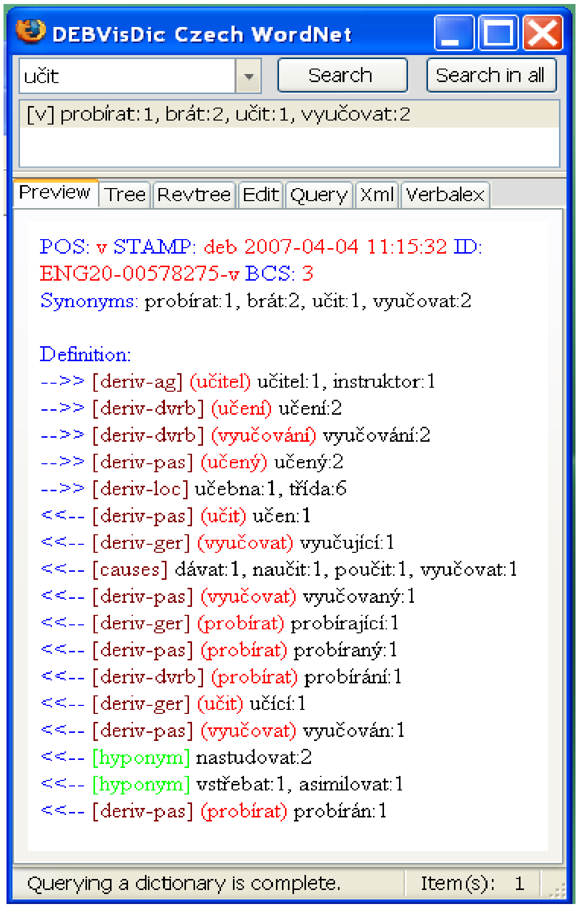
An example of the derivational analysis for Czech sufix *–ík*: it occurs with the nouns denoting agent or instrument (means), e.g. *zed-n-ík (bricklayer)* or *kapes-ník (hankerchief).*

First we want to derive agentive nouns: so we enter the suffix *–ík* and tag k1gM (noun, masculine animate) and generate the list of all words ending with *-ík*. The output is a list of 1210 nouns including proper names (from the original list of 126 000 Czech nouns). To obtain instrument nouns we input the tag k1gI (noun, masculine inanimate). As an output result we get a list of 715 nouns including proper names. The number of all words ending with suffix *-ík* (disregarding the grammatical tag) in stem dictionary of Ajka is 1830. The difference in the given numbers follows from the homonymy, for instance, some nouns can be both masculine animate and masculine inanimate (e.g. *náčelník* can denote – *chief* as well as *čelenka* – *headband*. Such cases have been checked manually.

In a similar way we have processed 22 Czech derivational suffixes and as a result we have obtained a detailed classification of the indicated derivations capturing agentive, instrumental, location and also resultative relations, for instance *spálit* -> *spálenina* (*to burn* -> *a burn*) which has not been mentioned before. At the same time the complete lists of all stems with the indicated suffixes together with labeling their semantic relations between the stems and respective suffixes was obtained as well. For the processed suffixes the coverage is complete (with regard to the list of 126 000 of the Czech noun stems).

Thus using the described procedure we are trying to find pairs of the word forms in which the first one is considered basic and the second one derived. The direction of the derivations is not always unambiguous but the most important goal is to establish the relation itself not its direction. The cases when changes in stem take place have to be checked and added manually.

2.4 D-relations in Czech and English WordNet



```
učit
[v] probírat:1, brát:2, učit:1, vyučovat:2

POS: v STAMP: deb 2007-04-04 11:15:32 ID:
ENG20-00578275-v BCS: 3
Synonyms: probírat:1, brát:2, učit:1, vyučovat:2

Definition:
-->> [deriv-ag] (učitel) učitel:1, instruktor:1
-->> [deriv-dvrb] (učení) učení:2
-->> [deriv-dvrb] (vyučování) vyučování:2
-->> [deriv-pas] (učený) učený:2
-->> [deriv-loc] učebna:1, třída:6
<<-- [deriv-pas] (učit) učen:1
<<-- [deriv-ger] (vyučovat) vyučující:1
<<-- [causes] dávat:1, naučit:1, poučit:1, vyučovat:1
<<-- [deriv-pas] (vyučovat) vyučovaný:1
<<-- [deriv-ger] (probírat) probírající:1
<<-- [deriv-pas] (probírat) probíraný:1
<<-- [deriv-dvrb] (probírat) probírání:1
<<-- [deriv-ger] (učit) učící:1
<<-- [deriv-pas] (vyučovat) vyučován:1
<<-- [hyponym] nastudovat:2
<<-- [hyponym] vstřebat:1, asimilovat:1
<<-- [deriv-pas] (probírat) probírán:1

Querying a dictionary is complete.    Item(s): 1
```

Figure1: D-relations in Czech WordNet

In Figure 1 we show how the D. relations are implemented in Czech Wordnet. As an example we show verbal synset {učit:1, vyučovat: probírat:1, brát:2}and the similar English one {teach:1, instruct:1}). It can be seen that there is a derivational subnet with five D-relations associated to {učit:1, ...} (in fact 14 but they are repeating with other literals in the synset as well). Each D-relation is labeled semantically so we have here the following D-relations: agentive, location, deverbative, gerund, passive – the last two may be characterized as more morphological (surface, see Sect. 2.1) than the first three.

In Princeton WordNet 3.0 we can observe the following three D-relations associated with the synset {teach:1, learn:5, instruct:1}

S: (v) **teach**, learn, instruct (impart skills or knowledge to) *"I taught them French"; "He instructed me in building a boat"*
derivationally related form

- W: (adj) teachable [Related to: teach] (ready and willing to be taught) *"docile pupils eager for instruction"; "teachable youngsters"*
- W: (n) teacher [Related to: teach] (a person whose occupation is teaching)
- W: (n) teacher [Related to: teach] (a personified abstraction that teaches) *"books were his teachers"; "experience is a demanding teacher"*
- W: (n) teaching [Related to: teach] (the activities of educating or instructing; activities that impart knowledge or skill) *"he received no formal education"; "our instruction was carefully programmed"; "good classroom teaching is seldom rewarded"*
- W: (adj) instructive [Related to: instruct] (serving to instruct or enlighten or inform)
- W: (n) instruction [Related to: instruct] (the activities of educating or instructing; activities that impart knowledge or skill) *"he received no formal education"; "our instruction was carefully programmed"; "good classroom teaching is seldom rewarded"*
- W: (n) instructor [Related to: instruct] (a person whose occupation is teaching)

It is not surprising that the full agreement between Czech and English D-relations includes only the agentive relation (*teach* -> *teacher*) and gerund relation (*teach* -> *teaching*). The relation *teach* -> *teachable* is not included among Czech relations at the moment but it will be easy to add it. The location relation is missing in English and also some others characterized usually as morphological. We included them in Czech WordNet – they belong to the set of the Czech derivational relations.

If we compare semantic labeling of the D-relations in both Wordnets we observe that they are more explicitly formulated in Czech Wordnet. The question that remains to be answered is how the different senses may be or are reflected in the individual derivations. In PWN 3.0 the derivation *teach – teacher* is given twice because there are two different senses of *teach* in PWN 3.0. In our view, it is enough to give this derivational relation just once because it is agentive in both cases. Of course, in Czech there are frequent cases like *držet* -> *držák* (*hold* -> *holder*) and *držet* -> *držitel* (*hold* -> *holder*) where the first one is instrument relation and the second agentive but in Czech the different suffixes have to be used (*-ák* vs. *-tel*) indicating a difference in gender as well (masculine inanimate vs. masculine animate).

3 What is the nature of the D-relations?

In the previous sections we have introduced the labeling of the Czech D-relations. The question may be asked what is the real nature of D-relations, whether it is semantic or rather morphological (formal). The D-relations exist between morphemes, typically between stems and corresponding suffixes. This formal feature makes them different from the relations between sentence constituents, as e.g. between verbs and their arguments. However, the main criterion for us is whether the particular relation affects meaning irrespective of its formal realization.

If we apply this criterion to the D-relations discussed above, such as deriv-ag, deriv-loc, deriv-instr, deriv-g, deriv-dem, deriv-pos, deriv-pro, we definitely come to the conclusion that their nature is semantic.

Then there are relations like deriv-an, deriv-na, deriv-dvrb, deriv-ger, deriv-aad, deriv-pas that are sometimes characterized as morphological only and their semantics is left aside. The first two relations hold between nouns and adjectives and both denote properties (e.g. deriv-an: *nový* -> *novost* (*new* -> *newness*)), but we have to take into account that there is something that may be called semantics of the parts of speech, i.e. in one case property is expressed by the adjective and then by the noun which is derived from the adjective. Deriv-na denotes property as well but here the adjective is derived from noun as in *boj -> bojovný* (*fight -> combative*). The relation deriv-dvrb exists between a verb and noun, e.g. *učit -> učení* (*teach -> teaching*), and it denotes action which is first expressed by the verb and then by the deverbative noun. We can say that in these cases the only difference lies in the optics of the individual parts of speech but this difference should be understood as semantic as well. However, it should be remarked once more that quite often the differences in the semantics of the parts of speech are not treated as truly semantic.

If we have look what standard Czech grammars (see e.g. Karlík et al, 1995) say about the semantics of the parts of speech we find the formulations such as: nouns denote independent entities, i.e. persons, animals and things and also properties and actions. Verbs then denote states and their changes and processes (actions) and their mutations. These descriptions certainly refer to the semantics of the nouns and verbs. They are usually followed by the explanations about morphological processes, i. e. usually derivations by which some parts of speech are formed from the others, as we have described them above. What is relevant and what is missing in the standard grammars are more detailed and extensive semantic classifications of nouns, verbs, as well as adjectives and numerals. They are beginning to appear only recently and have the form of ontologies – the standard grammars do not use this term at all.

Until we have such semantic classifications describing semantic relations between the individual parts of speech we can hardly have a full picture that is necessary for automatic processing of the derivational relations.

This issue certainly calls for a more detailed examination, which would be a topic for another paper.

4 The implementation of D-relations in Czech WordNet

The existing software tools (e.g.Visdic, cf. Horák, Smrž, 2004) used for building Wordnet databases standardly work with semantic relations between synsets and they treat them as atomic units. In fact, the synsets are not atomic as such and they consist of the smaller units called literals, i.e. for instance the synset {teach:1, instruct:1} contains two literals (lemmas).

If we want to deal with the D-relations automatically we immediately face a problem: because of their nature they typically hold not between synsets but between literals that as a rule belong to the different synsets, e.g. teach:1 and teacher:1. Therefore we need a tool that is able to define and create derivational links between the literals. According to our knowledge the only tool that can do this is DEBVisdic editor and browser developed at our NLP Lab at FI MU (cf. Horák, Pala, 2006, it can be downloaded from: http://nlp.-fi.muni.cz/projekty/deb2/clients/).

We have used it for the implementation of the D-relations in Czech WordNet (the result is shown in Sect. 2.4). The DEBVisdic tool is now used for representing and storing all the semantic relations including the D-relations. It is also exploited for building Wordnets in other languages such as Polish, Slovenian, Hungarian and others.

In our view, the way in which the D-relations (and other relations as well) are represented relevantly depends on the software tools used. This can be demonstrated if we compare the representation of the Czech D-relations in DEBVisdic with the one in PWN 3.0 (see Sect. 2.4) which appears to be less explicit and rather verbose. This also means that the representation used in PWN 3.0 will be probably less suitable for possible applications.

5 The results

As we said above after processing all D-relations by the derivational Ajka we have added the derived literals (lemmas) to the Czech WordNet. The final result – the number of the literals generated from the individual D-relations is given below together with their semantic labels:

deriv-na	641	(property, noun -> adj)
deriv-ger	1951	(property, verb -> adj)
deriv-dvrb	5041	(action, verb -> noun)
deriv-pos	4073	(possessive, noun -> adj)
deriv-pas	9801	(passive, verb -> adj)
deriv-aad	1416	(property, adj -> adverb)

deriv-an ………...1930 (property, adj -> noun)

deriv-g …………2695 (gender, noun -> noun)

deriv-ag …………186 (agentive, verb -> noun)

deriv-dem ………3695 (diminutive, noun -> noun)

Total ………… 31429 literals

These numbers also tell us how productive the particular relations are. Note that the most frequent is passive relation which is followed by the deverbative (action) relation. The third most frequent relation is a possessive one. It would be interesting to examine what these facts can tell us about semantic structure of texts.

6 Conclusions

In the paper we present the first results of computational analysis of the basic and most regular D-relations in Czech using derivational version of the morphological analyzer Ajka.

Though the analysis is far from complete at the moment the number of the generated items has led us to the decision to include them in Czech WordNet and enrich it considerably with the derivational nests (subnets). In our view, this kind of enrichment makes Czech WordNet more suitable for some applications, namely for searching.

The second and even more important reason for doing all this is a belief that the derivational relations and derivational subnets created by them reflect basic cognitive structures existing in natural language. More effort is needed for exploring them from the point of view of now so popular ontologies – they certainly offer a formal ground (they are expressed by the individual morphemes) for natural language based ontologies.

We have also included a brief comparison with the recently released Princeton WordNet 3.0 which now contains derivational links for English as well. As we expected the comparison confirms the known fact that English as an analytic language is much poorer with regard to the derivational relations than the inflectional ones.

From the technical point of view PWN 3.0 is still not using the representation in XML format (as DebVisdic does) and this, we think, in certain degree limits the possibilities to express some of the links in a standard way. The present web interface where Princeton WordNet 3.0 can be browsed: http://wordnet.princeton.edu/perl/webwn) does not seem to be able to work directly with the links between literals.

On the other hand, we are well aware that adding D-relations to PWN 3.0 is very stimulating and useful though it will be quite demanding to establish the derivational links between English and other languages (through Interlingual Index). This makes it a new challenge for the whole WordNet community.

Acknowledgements

The research was supported by the grant projects GA 201/05/2871, 1ET100300419 and NPVII 2C06009.

References

Horák A., Pala K., Rambousek A., and Povolný M. 2006. First version of new client-server wordnet browsing and editing tool. In *Proceedings of the Third International WordNet Conference – GWC 2006*, p. 325-328, Jeju, South Korea, Masaryk University, Brno.

Horák A., Smrž P. 2004. Visdic – WordNet Editing and Browsing Tool, Proceedings of the 2nd GWC, Brno, Masaryk University.

Karlík P. et al. 1995, Příruční mluvnice češtiny (Every day Czech Grammar), Nakladatelství Lidové Noviny, Prague, pp. 229, 310.

Pala K., Sedláček R., Veber M. 2003. Relations between Inflectional and Derivation Patterns, Proceedings of EACL, Budapest.

Petr J. et al. 1986. *Mluvnice češtiny 1*, Praha: Academia.

Sedláček R., Smrž P. 2001. A New Czech Morphological Analyser Ajka. Proceedings of the 4th International Conference on Text, Speech and Dialogue, Springer Verlag, Berlin, s.100-107.

Vossen P. 2003. EuroWordNet General Document, Version 3, University of Amsterdam.

Web address of the Princeton WordNet 3.0: http://wordnet.princeton.edu/perl/webwn.

Multilingual Word Sense Discrimination: A Comparative Cross-Linguistic Study

Alla Rozovskaya
Department of Linguistics
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801
rozovska@uiuc.edu

Richard Sproat
Department of Linguistics
Univ. of Illinois at Urbana-Champaign
Urbana, IL 61801
rws@uiuc.edu

Abstract

We describe a study that evaluates an approach to Word Sense Discrimination on three languages with different linguistic structures, English, Hebrew, and Russian. The goal of the study is to determine whether there are significant performance differences for the languages and to identify language-specific problems. The algorithm is tested on semantically ambiguous words using data from Wikipedia, an online encyclopedia. We evaluate the induced clusters against sense clusters created manually. The results suggest a correlation between the algorithm's performance and morphological complexity of the language. In particular, we obtain FScores of 0.68 , 0.66 and 0.61 for English, Hebrew, and Russian, respectively. Moreover, we perform an experiment on Russian, in which the context terms are lemmatized. The lemma-based approach significantly improves the results over the word-based approach, by increasing the FScore by 16%. This result demonstrates the importance of morphological analysis for the task for morphologically rich languages like Russian.

1 Introduction

Ambiguity is pervasive in natural languages and creates an additional challenge for Natural Language applications. Determining the sense of an ambiguous word in a given context may benefit many NLP tasks, such as Machine Translation, Question Answering, or Text-to-Speech synthesis.

The *Word Sense Discrimination* (WSD) or *Word Sense Induction* task consists of grouping together the occurrences of a semantically ambiguous term according to its senses. Word Sense Discrimination is similar to Word Sense Disambiguation, but allows for a more unsupervised approach to the problem, since it does not require a pre-defined set of senses. This is important, given the number of potentially ambiguous words in a language. Moreover, labeling an occurrence with its sense is not always necessary. For example, in Information Retrieval WSD would be useful for the identification of documents relevant to a query containing an ambiguous term.

Different approaches to WSD have been proposed, but the evaluation is often conducted using a single language, so it is difficult to predict performance on another language. To the best of our knowledge, there has not been a systematic comparative analysis of WSD systems on different languages. Yet, it is interesting to see whether there are significant differences in performance when a method is applied to several languages that have different linguistic structures. Identifying the reasons for performance differences might suggest what features are useful for the task.

The present project adopts an approach to WSD that is based on similarity measure between context terms of an ambiguous word. We compare the performance of an algorithm for WSD on English, Hebrew, and Russian, using lexically ambiguous words and corpora of similar sizes.

We believe that testing on the above languages

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 82–87,
Prague, June 2007.

might give an idea about how accuracy of an algorithm for WSD is affected by language choice. Russian is a member of the Slavic language group and is morphologically rich. Verbs, nouns, and adjectives are characterized by a developed inflectional system, which results in a large number of wordforms. Hebrew is a Semitic language, and is complex in a different way. In addition to the root-pattern morphology that affects the word stem, it also has a complex verb declination system. Moreover, function words, such as prepositions and determiners, cliticize, thereby increasing the number of wordforms. Lastly, cliticization, coupled with the absence of short vowels in text, introduces an additional level of ambiguity for Hebrew.

There are two main findings to this study. First, we show that the morphological complexity of the language affects the performance of the algorithm for WSD. Second, the lemma-based approach to Russian WSD significantly improves the results over the word-based approach.

The rest of the paper is structured as follows: first, we describe previous work that is related to the project. Section 3 provides details about the algorithm for WSD that we use. We then describe the experiments and the evaluation methodology in Sections 4 and 5, respectively. We conclude with a discussion of the results and directions for future work.

2 Related Work

First, we describe several approaches to WSD that are most relevant to the present project: Since we are dealing with languages that do not have many linguistic resources available, we chose a most unsupervised, knowledge-poor approach to the task that relies on words occurring in the context of an ambiguous word. Next, we consider two papers on WSD that provide evaluation for two languages. Finally, we describe work that is concerned with the role of morphology for the task.

2.1 Approaches to Word Sense Discrimination

Pantel and Lin (2002) learn word sense induction from an untagged corpus by finding the set of the most similar words to the target and by clustering the words. Each word cluster corresponds to a sense. Thus, senses are viewed as clusters of words.

Another approach is based on clustering the occurrences of an ambiguous word in a corpus into clusters that correspond to distinct senses of the word. Based on this approach, a sense is defined as a cluster of contexts of an ambiguous word. Each occurrence of an ambiguous word is represented as a vector of features, where features are based on terms occurring in the context of the target word. For example, Pedersen and Bruce (1997) cluster the occurrences of an ambiguous word by constructing a vector of terms occurring in the context of the target. Schütze (1992) presents a method that explores the similarity between the context terms occurring around the target. This is accomplished by considering feature vectors of context terms of the ambiguous word. The algorithm is evaluated on natural and artificially-constructed ambiguous English words.

Sproat and van Santen (1998) introduce a technique for automatic detection of ambiguous words in a corpus and measuring their degree of polysemy. This technique employs a similarity measure between the context terms similar in spirit to the one in (Schütze, 1992) and singular value decomposition in order to detect context terms that are important for disambiguating the target. They show that the method is capable of identifying polysemous English words.

2.2 Cross-Linguistic Study of WSD

Levinson (1999) presents an approach to WSD that is evaluated on English and Hebrew. He finds 50 most similar words to the target and clusters them into groups, the number of groups being the number of senses. He reports comparable results for the two languages, but he uses both morphologically and lexically ambiguous words. Moreover, the evaluation methodology focuses on the success of disambiguation for an ambiguous word, and reports the number of ambiguous words that were disambiguated successfully.

Davidov and Rappoport (2006) describe an algorithm for unsupervised discovery of word categories and evaluate it on Russian and English corpora. However, the focus of their work is on the discovery of semantic categories and from the results they report for the two languages it is difficult to infer how the languages compare against each other.

We conduct a more thorough evaluation. We also control cross-linguistically for number of training examples and level of ambiguity of selected words, as described in Section 4.

2.3 Morphology and WSD

McRoy (1992) describes a study of different sources useful for word sense disambiguation, including morphological information. She reports that morphology is useful, but the focus is on derivational morphology of the English language. In the present context, we are interested in the effect of inflectional morphology on WSD, especially for languages, such as Russian and Hebrew.

Gaustad (2004) proposes a lemma-based approach to a Maximum Entropy Word Sense Disambiguation System for Dutch. She shows that collapsing wordforms of an ambiguous word yields a more robust classifier due to the availability of more training data. The results indicate an improvement of this approach over classification based on wordforms.

3 Approach

Our algorithm relies on the method for selection of relevant contextual terms and on distance measure between them introduced in (Sproat and van Santen, 1998) and on the approach described in (Schütze, 1998), though the details of clustering differ slightly. The intuition behind the algorithm can be summarized as follows: (1) words that occur in the context of the ambiguous word are useful for determining its sense; and (2) contextual terms of an ambiguous word belong to topics corresponding to the senses of the ambiguous word. Before describing the algorithm in detail, we give an overview of the system.

The algorithm starts by collecting all the occurrences of an ambiguous word in the corpus together with the surrounding context. Next, we build a symmetric distance matrix D, where rows and columns correspond to context terms, and D[i][j] is the distance value of term *i* and term *j*. The distance measure is supposed to reflect how the two terms are close semantically (whether they are related to the same topic). For example, we would expect the distance between the words *financial* and *money* to be smaller than the distance between the words *financial* and *river*: The first pair is more likely to occur in the same context, than the second one. Using the distance measure, the context terms are partitioned into sense clusters. Finally, we group the sentences containing the ambiguous word into sentence clusters using the context term clusters.
We now describe each step in detail:

1. We collect contextual terms of an ambiguous word *w* in a context window of 50 words around the target. Each context term *t* is assigned a *weight* (Sproat and J. van Santen, 1998):

$$w_t = \frac{CO(t|w)}{FREQ(t)} \quad (1)$$

   $CO(t|w)$ is the frequency of the term in the context of w, and $FREQ(t)$ is the frequency of the term in the corpus. Term weights are used to select context terms that will be helpful in determining the sense of the ambiguous word in a particular context. Furthermore, term weights are employed in (4) in sentence clustering.

2. For each pair t_i and t_j of context terms, we compute the distance between them (Sproat and J. van Santen,1998):

$$D_w[i][j] = 1 - \frac{[\frac{CO_w(t_i|t_j)}{FREG(t_i)} + \frac{CO_w(t_j|t_i)}{FREQ(t_j)}]}{2} \quad (2)$$

   $CO_w(t_i|t_j)$ is the frequency of t_i in the context of t_j, and $FREQ(t_i)$ is the frequency of t_i in the training corpus. We assume that the distance between t_i and t_j is inversely proportional to the semantic similarity between t_i and t_j.

3. Using the distance matrix from (2), the context terms are clustered using an agglomerative clustering technique:

   - Start by assigning each context term to a separate cluster
   - While stopping criterion is false: merge two clusters whose distance [1] is the smallest.[2]

[1] There are several ways to define the distance between clusters. Having experimented with three - Single Link, Complete Link and Group Average, it was found that Complete Link definition works best for the present task. (*Complete Link* distance between clusters *i* and *j* is defined as the maximum distance between a term from cluster *i* and a term from cluster *j*).

[2] In the present study, the clusters are merged as long as the

The output of step (3) is a set of context term clusters for the target word. Below are shown select members for term clusters for the English word *bass*:

Cluster 1: songwriter singer joined keyboardist

Cluster 2: waters fishing trout feet largemouth

4. Finally, the sentences containing the ambiguous word are grouped using the context term clusters from (3). Specifically, given a sentence with the ambiguous word, we compute the score of the sentence with respect to each context word cluster in (3) and assign the sentence to the cluster with the highest score. The score of the sentence with respect to cluster c is the sum of weights of sentence context terms that are in c.

4 Experiments

The algorithm is evaluated on 9 ambiguous words with two-sense distinctions. We select words that (i) have the same two-sense distinction in all three languages or (ii) are ambiguous in one of the languages, but each of their senses corresponds to an unambiguous translation in the other two languages. In the latter case, the translations are merged together to create an artificially ambiguous word. We believe that this selection approach allows for a collection of a comparable set of ambiguous words for the three languages. An example of an ambiguous word is the English word *table*, that corresponds to two gross sense distinctions (*tabular array*, and *a piece of furniture*). This word has two translations into Russian and Hebrew, that correspond to the two senses. The selected words are presented in Table 1.

The words display different types of ambiguity. In particular, disambiguating the Hebrew word *gishah* (access; approach) or the Russian word *mir* (peace; world) would be useful in Machine Translation, while determining the sense of a word like *language* would benefit an Information Retrieval system. It should also be noted that several words possess additional senses, which were ignored because they rarely occurred in the corpus. For example, the Russian word *yazyk* (language) also has the meaning of *tongue* (body part).

The corpus for each language consists of 15M word tokens, and for the same ambiguous word the same number of training examples is selected from each language. For each ambiguous word, a set of 100-150 examples together with 50 words of context is selected from the section of the corpus not used for training. These examples are manually annotated for senses and used as the test set for each language.

5 Evaluation Methodology

The evaluation is conducted by comparing the induced sentence clusters to clusters created manually. We use three evaluation measures : *cluster purity*, *entropy*, and *FScore*. [3]

For a cluster C_r of size q_r, where the size is the number of examples in that cluster, the dominating sense S_i in that cluster is selected and *cluster purity* is computed as follows:

$$P(C_r) = \frac{n_r^i}{q_r}, \qquad (3)$$

where n_r^i is the number of examples in cluster C_r with sense S_i.

For an ambiguous word *w*, cluster purity *P(w)* is the weighted average of purities of the clusters for that word. [4]. Higher cluster purity score corresponds to a better clustering outcome.

Entropy and *FScore* measures are described in detail in Zhao and Karypis (2005). *Entropy* indicates how distinct senses are distributed between the two clusters. The perfect distribution is the assignment of all examples with sense 1 to one cluster and all examples with sense 2 to the other cluster. In such case, the entropy is 0. In general, a lower value indicates a better cluster quality. Entropy is computed for each cluster. Entropy for word *w* is the weighted average of the entropies of the clusters for that word.

Finally, *FScore* considers both the coverage of the algorithm and its ability to discriminate between the two senses. *FScore* is computed as the harmonic

number of clusters exceeds the number of senses of the ambiguous word in the test data.

[3] Examples whose scores with respect to all clusters are zero (examples that do not contain any terms found in the distance matrix) are not assigned to any cluster, and thus do not affect cluster purity and cluster entropy. This is captured by the FScore measure described below.

[4] In the present study, the number of clusters and the number of senses for a word is always 2

Senses	English	Hebrew	Russian
access;approach	access;approach	gishah	dostup;podxod
actor;player	actor;player	saxqan	akter;igrok
evidence; quarrel	argument	vikuax;nimuq	argument
body part; chief	head	rosh	golova;glava
world;peace	world; peace	shalom;olam	mir
furniture; tabular array	table	shulxan;tavlah	stol;tablitza
allow;resolve	allow;resolve	hershah;patar	razreshat'
ambiance; air	atmosphere	avira;atmosfera	atmosfera
human lang.;program. lang.	language	safah	yazyk

Table 1: Ambiguous words for testing: The first column indicates the senses; unambiguous translations that were merged to create an ambiguous word are indicated by a semicolon

mean of *Precision* and *Recall*, where recall and precision for sense S_i with respect to cluster C_r are computed by treating cluster C_r as the output of a retrieval system for sense S_i .

6 Results and Discussion

We show results for two experiments. Experiment 1 compares the algorithm's performance cross-linguistically without morphological analysis applied to any of the languages. Experiment 2 compares the performance for Russian in two settings: with and without morphological processing performed on the context terms.

Table 2 presents experimental results. Baseline is computed by assigning the most common sense to all occurrences of the ambiguous word. We observe that English achieves the highest performance both in terms of cluster purity and FScore, while Russian performs most poorly among the three languages. This behavior may be correlated with the average frequency of the context terms that are used to construct the distance matrix in the corpus (cf. 7 for English and 4.2 for Russian). In particular, the difference in the frequencies can be attributed to the morphological complexity of Russian, as compared to English and Hebrew. Hebrew is more complex than English morphologically, which would account for a drop in performance for the Hebrew words vs. the English words. Furthermore, one would expect a higher degree of ambiguity for Hebrew due to the absence of short vowels in text.

It is worth noting that while both Hebrew and Russian possess features that might negatively affect the performance, Hebrew performs better than Russian. We hypothesize that cliticization and the lack of vowels in text are not as significant factors for the performance as the high inflectional nature of a language, such as Russian. We observe that the majotity of the context terms selected by the algorithm for disambiguation belong to the noun category. This seems intuitive, since nouns generally provide more information content than other parts of speech and thus should be useful for resolving lexical ambiguity. While an English or a Hebrew noun only has several wordforms, a Russian noun may have up to 12 different forms due to various inflections.

The morphological complexity of Russian affects the performance in two ways. First, cluster purity is affected, since the counts of context terms are not sufficiently reliable to accurately estimate term distances. Incorrect term distances subsequently affect the quality of the term clusters. Second, the percentage of default occurrences (examples that have no context terms occurring in the distance matrix) is the least for English (0.22) and the highest for Russian (0.27). The default occurrences affect the recall.

The results of experiment 2 support the fact that morphological complexity of a language negatively affects the performance. In that experiment, the inflections are removed from all the context terms. We apply a morphological analyzer [5] to the corpus and replace each word with its lemma. In 10% of the word tokens, the analyzer gives more than one possible analysis, in which case the first analysis is selected. As can be seen in Table 2 (last row), removing inflections produces a significant improvement both in recall and precision, while preserving the cluster purity and slightly reducing cluster entropy. Moreover, the performance in terms of recall, precision, and coverage is better than for English and

[5] Available at *http://www.aot.ru/*

Language	Baseline	Coverage	Precision	Recall	FScore	Purity	Entropy
English	0.73	0.78	0.77	0.61	0.68	0.79	0.61
Hebrew	0.72	0.79	0.76	0.58	0.66	0.82	0.59
Russian	0.72	0.73	0.70	0.54	0.61	0.81	0.62
Russian(lemma)	0.72	0.80	0.77	0.66	0.71	0.82	0.61

Table 2: Results: Baseline is the most frequent sense; coverage is the number of occurrences on which the decision was made by the algorithm

Hebrew.

7 Conclusions and Future Work

We have described a cross-linguistic study of a Word Sense Discrimination technique. An algorithm based on context term clustering was applied to ambiguous words from English, Hebrew, and Russian, and a comparative analysis of the results was presented. Several observations can be made. First, the results suggest that the performance can be affected by morphological complexity in the case of a language, such as Russian, specifically, both in terms of precision and recall. Second, removing inflectional morphology not only boosts the recall, but significantly improves the precision. These results support the view that morphological processing is beneficial for WSD.

For future work, we plan to investigate more thoroughly the role of morphological analysis for WSD in Russian and Hebrew. In particular, we will focus on the inflectional morphology of Russian in order to determine whether removing inflections consistently improves results for Russian ambiguous words across different parts of speech. Further, considering the complex structure of the Hebrew language, we would like to determine what kind of linguistic processing is useful for Hebrew in the WSD context.

Acknowledgments

We are grateful to Roxana Girju and the anonymous reviewers for very useful suggestions and comments. This work is funded in part by grants from the National Security Agency and the National Science Foundation.

References

Dmitry Davidov and Ari Rappoport 2006. Efficient Unsupervised Discovery of Word Categories Using Symmetric Patterns and High Frequency Words. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 297–304. Sydney, Australia.

Richard O. Duda and Peter E. Hart. 1973. *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York.

Tanja Gaustad. 2004. A Lemma-Based Approach to a Maximum Entropy Word Sense Disambiguation System for Dutch. In *Proceedings of the 20th International Conference on Computational Linguistics (Coling 2004)*, 778-784. Geneva.

Dmitry Levinson. 1999. Corpus-Based Method for Unsupervised Word Sense Disambiguation. *www.stanford.edu/ dmitryle/acai99w1.ps*.

Susan Weber McRoy. 1992. Using Multiple Knowledge Sources for Word Sense Discrimination. *Computational Linguistics*, 18(1): 1–30.

Patrick Pantel and Dekang Lin. 2002. Discovering Word Senses from Text In *In Proceedings of ACM SIGKDD*, pages 613-619. Edmonton.

Ted Pedersen and Rebecca Bruce. 1997. Distinguishing word senses in untagged text. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, 197-207. Providence, RI, August.

Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123.

Richard Sproat and Jan van Santen. 1998. Automatic ambiguity detection. In *Proceedings of International Conference on Spoken Language Processing* . Sydney, Australia, 1998.

Ying Zhao and George Karypis. 2005. Hierarchical Clustering Algorithms for Document Datasets *Data Mining and Knowledge Discovery*, 10(2):141–168.

Named Entity Recognition for Ukrainian: A Resource-Light Approach

Sophia Katrenko
HCSL, University of Amsterdam,
Kruislaan 419, 1098VA Amsterdam,
the Netherlands
katrenko@science.uva.nl

Pieter Adriaans
HCSL, University of Amsterdam,
Kruislaan 419, 1098VA Amsterdam,
the Netherlands
pitera@science.uva.nl

Abstract

Named entity recognition (NER) is a subtask of information extraction (IE) which can be used further on for different purposes. In this paper, we discuss named entity recognition for Ukrainian language, which is a Slavonic language with a rich morphology. The approach we follow uses a restricted number of features. We show that it is feasible to boost performance by considering several heuristics and patterns acquired from the Web data.

1 Introduction

The information extraction task has proved to be difficult for a variety of domains (Riloff, 1995). The extracted information can further be used for question answering, information retrieval and other applications. Depending on the final purpose, the extracted information can be of different type, e.g., temporal events, locations, etc. The information corresponding to locations and names, is referred to as the information about named entities. Hence, named entity recognition constitutes a subtask of the information extraction in general.

It is especially challenging to extract the named entities from the text sources written in languages other than English which, in practice, is supported by the results of the shared tasks on the named entity recognition (Tjong Kim Sang, 2002).

Named entity recognition for the languages with a rich morphology and a free word order is difficult because of several reasons. The entropy of texts in such languages is usually higher than the entropy of English texts. It is either needed to use such resources as morphological analyzers to reduce the data sparseness or to annotate a large amount of data in order to obtain a good performance. Luckily, the free word order is not crucial for the named entity recognition task as the local context of a named entity should be sufficient for its detection. Besides, a free word order usually implies a free order of constituents (such as noun phrases or verb phrases) rather than words as such. For instance, although (1)[1] is grammatically correct and can occur in the data, it would be less frequent than (2).

(1) червону вона тримає квітку
red she holds flower
she holds a red flower

(2) вона тримає червону квітку
she holds red flower
she holds a red flower

The first phrase exemplifies that an adjective 'червону' is in a focus, whereas the second reflects the word order which is more likely to occur. In terms of named entities, an entity consisting of several words is also less likely to be split (consider, e.g., *National saw she Bank* where 'National Bank' represents one named entity of type organization). In the newspaper corpus we annotated, we have observed no examples of split named entities.

In this paper, we study different data representation and machine learning methods to extract the named entities from text. Our goal is two-fold. First,

[1] all examples in the paper are in Ukrainian, for convenience translated and sometimes transliterated

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 88–93,
Prague, June 2007. ©2007 Association for Computational Linguistics

we explore the possibility of using patterns induced from the data gathered on the Web. We also consider Levenshtein distance to find the most similar instances in the test data given a training set. Besides, we study the impact of different feature sets on the resulting classification performance.

We start with the short overview of the methods for NER proposed by the IE community. Afterwards, the experiments are described. We conclude with the outlook for the future work.

2 Related work

The existing NER systems use many sources in order to be able to extract NEs from the text data. Some of them rely on hand-written rules and precompiled lists of city names, person names and other NEs in a given language, while others explore methods to automatically extract NEs without prior knowledge. In the first case, the gazetteers will in most cases improve NER results (Carreras et al., 2002) but, unfortunately, they may not exist for a language one is working on. Hand-written rules can also cover more NEs but building such patterns will be a very time-consuming process.

There have been many methods applied to NER, varying from the statistical to the memory-based approaches. Most work on NER has been focused on English but there are also approaches to other languages such as Spanish (Kozareva et al., 2005), German, or Dutch. In addition, several competitions have been organized, with a focus on multilingual NER (Tjong Kim Sang, 2002). While analyzing the results of these shared tasks, it can be concluded that the selected features are of a great importance. In our view, they can be categorized in two types, i.e. *contextual* and *orthographic* [2]. The first type includes words surrounding a given word while the other contains such features as capitalized letters, digits contained within the word, etc. Both types of features contribute to the information extraction task. Nevertheless, orthographic features can already be language-specific. For instance, capitalization is certainly very important for such languages as English or Dutch but it might be less useful for German.

The feature set of some NER methods (Wu, 2002) also includes part-of-speech information and/or word prefixes and suffixes. Although this information (and especially lemmas) is very useful for the languages with rich morphology, it presupposes the existence of POS taggers for a given language.

Another conclusion which can be drawn relates to the machine learning approaches. The best results have been received by applying ensemble methods (Wu, 2002; Florian, 2002; Carreras et al., 2002).

A very interesting work on named entity recognition task was reported by Collins et al. (1999) who used only few named entities to bootstrap more. The other approach proposed recently makes use of the data extracted from the Web (Talukdar et al., 2006). By restricting themselves to the fixed context of the extracted named entities and by employing grammar inference techniques, the authors filter out the most useful patterns. As they show, by applying such approach precision can already be largely boosted.

Pastra et al. (2002) focused on the applicability of already existing resources in one language to another. Their case study was based on English and Romanian, where a system, originally developed for NER in English was adapted to Romanian. Their results suggest that such adaptation is easier than developing a named entity recognition system for Romanian from scratch. However, the authors also mention that not all phenomena in Romanian have been taken into account which resulted in low recall.

3 Methodology

Ukrainian belongs to the languages where the named entities are usually capitalized, which makes their detection relatively easy. In this paper we focus on using minimal information about the language in combination with the patterns learnt from the Web data, features extracted from the corpus and Levenshtein similarity measure.

Our hypothesis behind all three components is the following. We expect orthographic features be useful for a named entity detection but not sufficient for its classification. Contextual information may already help but as we do not intend to use lemmas but words instead, it will likely not boost recall of the named entity recongnition. To be able to detect more named enities in the text, we propose to use pat-

[2] Sometimes, these types of features are referred to as *word-external* and *word-internal* (Klein et al., 2003)

terns collected from the Web and Levenshtein similarity measure. Patterns from the Web should provide more contextual information than can be found in a corpus. In addition, a similarity measure gives us an opportunity to detect the named entities which have the same stem. The latter is especially useful when the same entity was mentioned in the training set as well as in the test data but its flections differ.

The intention of our study is, therefore, to start with a standard set of features (contextual and orthographic) as used for the many languages in the past and to add some means which would account for the fact that Ukrainian is a highly-inflected language.

3.1 Classification

First, we consider the features which can be easily extracted given the data, such as contextual and orthographic ones as described below in Table 1. For each word in a corpus its context (2 tokens to left and to the right) and its orthographic features are extracted. Orthographic features are binary features which, for instance, indicate whether a word is capitalized (1 or 0), etc. We have selected the following machine learning methods: k-nearest neighbor (knn) and voting and stacking as the ensemble methods which have been successfully applied to the named entity recognition task in the past.

contextual	-2/+2 words
	orthographic
CAP	capitalized
ALLCAP	all elements of a token capitalized
BSENT	first token in a sentence
NUM	contains digits
QUOTE	contains quotes

Table 1: Features

To overcome data sparseness and to increase recall, we make use of two techniques. First, we apply the patterns extracted from the Web.

3.2 Patterns

If we wish to collect patterns for a certain category C of the named entities (e.g.), we first collect all named entities that fall into it. Then, for each $X \in C$, we use X as a query term for Google (for this purpose we used the Google API). The queries we constructed were mainly based on the locations, such as 'Київ', 'Львів', 'Харків', 'Чернівці' etc. For each of these words we created queries by declining them (as there are 7 cases in Ukrainian language which causes the high variability). Consequently, we get many snippets where X occurs. To extract patterns from snippets, we fix a context and use 2 words to the left and to the right of X as in the classification approach above. The patterns which only consist of a named entity, closed-class words (e.g., prepositions, conjunctions, etc.) and punctuation are removed as such that do not provide enough evidence to classify an instance.

Intuitively, if there are many patterns acquired from the large collection of data on the Web, they must be sufficient (in some sense even redundant) to recognize named entities in a text. For instance, such pattern as *was located in* X in English can correspond to three patterns in Ukrainian *was located (fem., sing.) in* X, *was located (masc., sing.) in* X, *was located (neut., sing.) in* X. Even though these patterns could be embraced in one, we are rather interested in collecting all possible patterns avoiding this way stemming and morphological analysis.

As in Talukdar's approach (Talukdar et al., 2006), we expect patterns to provide high precision. We are, however, concerned about the size of Ukrainian Web which is much smaller than English part of the Web. As a consequence, it is not clear whether recall can be improved much by using the Web data.

3.3 Levenshtein distance

Yet another approach to address rich morphology of Ukrainian, is to carry out a matching of probable named entities in a test set against a list of named entities in a training set. It can be done by using string edit distances, such as Levenshtein.

Levenshtein (or edit) distance of two strings, x and y is measured as the minimal number of insertions, deletions, or substitutions to transform one string into the other. Levenshtein distance has become popular in the natural language processing field and was used for the variety of tasks (e.g., semantic role labeling).

Definition 1 (Levenshtein distance) *Given two sequences $x = x_1x_2 \dots x_n$ and $y = y_1y_2 \dots y_m$ of a length n and m respectively, Levenshtein distance is defined as follows*

$$lev(i,j) = min \begin{cases} lev(i-1,j-1) + d(x_i,y_j) \\ lev(i-1,j) + 1 \\ lev(i,j-1) + 1 \end{cases}$$

In the definition above, $d(x_i, y_j)$ is a cost of substituting one symbol in x by a symbol from y. The insertion and deletion costs are equal to 1.

Let $\mathcal{A}$ be a candidate named entity and $\mathcal{L}$ a list of all named entities found in the training set. By computing the Levenshtein distance between $\mathcal{A}$ and each element from $\mathcal{L}$, the nearest neighbor to $\mathcal{A}$ will be a NE with the lowest Levenshtein score. It might, however, happen that there are no named entities in a training set that correspond to the candidate in a test set. Consider, for instance the Levenshtein distance of two words 'Юрій' (George) and 'Крім' (besides) which is equal to 2. Even though the distance is low, we do not wish to classify 'Крім' as a named entity whose type is PERSON because it is simply a preposition. The problem we described can be solved in several ways. On the one hand, it is possible to use a list of stop words with most frequent prepositions, conjunctions and pronouns listed. On the other hand, we can also set a threshold for the Levenshtein distance. In the experiments we present below, we avoid setting threshold by using a simple heuristics. We align the first letters of $\mathcal{A}$ with its nearest neighbor. If they do not match (as in example above), we conclude that no variants of $\mathcal{A}$ belong to the training set.

4 Experiments and Evaluation

We have conducted three types of experiments using different feature sets, patterns extracted from the Web and Levenshtein distance. We expect that both types of experiments can shed a light on usefulness of the features that we defined for NER on Ukrainian data.

4.1 Data

Initially, several articles of the newspaper Mirror Weekly (year 2005)[3] were annotated. During the annotating process we considered the following named instances: PERSON (person names), LOC (location), ORG (organization).In total, there were 10,000 tokens annotated, 514 of which are named entities. All named entities have been annotated according to the IOB annotation scheme (Ramshaw and Marcus, 1995). The annotated corpus can be downloaded from `http://www.science.uva.nl/~katrenko/Corpus`

The corpus was split into training and test sets of 6,606 and 3,397 tokens, respectively. The corpus is relatively small but we hope to study whether such features as orthographic are sufficient for the NER task alone or it is needed to add more sources to approach this task.

4.2 Classification

The results of our experiments on classification of named entities are provided in Table 2. Baseline B_1 was defined by the most frequent tag in the data (ORG). Similarly to Conll shared task (Tjong Kim Sang, 2002), we also calculated a baseline by tagging all named entities which occurred in the training set (B_2). Although there are many names of organizations detected, there are only 1,92% of person names recognized.

	precision	recall	F-score
B_1	0.32	0.32	0.32
B_2	0.29	0.18	0.22
M^{2-knn}_{ortho}	0.31	0.44	0.36
$M^{2-knn}_{ortho+cont}$	0.38	0.46	0.42
$M^{Voting}_{ortho+cont}$	0.47	0.38	0.42
$M^{Stacking}_{ortho+cont}$	0.40	0.43	0.41
$M^{Voting}_{ortho+cont+pat}$	0.46	0.39	0.42
$M^{Voting}_{ortho+cont+pat+lev}$	**0.50**	**0.46**	**0.48**

Table 2: Experiments: precision and recall

Since we are interested in how much each type of the feature sets contributes to the classification accuracy, we have conducted experiments on contextual features only, on orthographic features only (model M^{2-knn}_{ortho} in Table 2) and on the combinations of both (model $M^{2-knn}_{ortho+cont}$ in Table 2). When used alone, contextual features do not provide a high performance. However, their combination with the orthographic features already results in a higher precision (at expense of recall) and in a higher F-score. It is worth noting that all results given in Table 2 were obtained either by using memory-based learning (in particular, k-nearest neighbor as in M^{2-knn}_{ortho} and in $M^{2-knn}_{ortho+cont}$) or by ensemble methods (as in $M^{Voting}_{ortho+cont}$ and $M^{Stacking}_{ortho+cont}$). The latter option was particularly interesting to explore because it proved to provide accurate results for the

[3] can be found at `http://www.zn.kiev.ua`

named entity recognition task in the past. The results in Table 2 also seem to support a claim that the ensemble methods perform better. It can be seen when comparing $M^{2-knn}_{ortho+cont}$, $M^{Voting}_{ortho+cont}$ and $M^{Stacking}_{ortho+cont}$. Despite of using the same feature sets, Voting (based on Naive Bayes, decision trees and 2-knn) and Stacking (2-knn as a meta-learner applied to Naive Bayes and decision tree learner) both provide higher precision but lower recall.

By using χ^2 test on the training set, we determined which attributes are the most informative for the classification task. The most informative turned out to be a word itself followed by the surrounding context (one token to the right and to the left). The least informative feature is NUM, apparently because there have been not many named entities containing digits.

4.3 Patterns

As a next step, we employed the patterns extracted from the Web data. Some of the patterns accompanied with the translation and information on case are given in Table 3. It can be noticed that not all of the patterns are accurate. For instance, a pattern *together with a city mayor LOC* can also be used to extract a name of a mayor (hence, PERSON) and not a location (LOC). Patterns containing prepositions (so, mostly patterns containing a named entity in locative case) 'in', 'with', 'nearby' are usually more accurate but they still require additional context (as a word 'town' in *in a little town LOC*).

The results we obtained by employing such patterns did not significantly change the overall performance (Table 2, model $M^{Voting}_{ortho+cont+pat}$). However, the performance on some categories such as ORG or LOC (Table 5 and Table 6, model *ALL+P*) was positively affected in terms of F-score.

4.4 Levenshtein distance

Finally, we compare all capitalized words in a test set against the named entities found in the training data. The first 6 examples in Table 4 show the same nouns but in different cases. The distance in each case is equal 1. Since we did not carry out the morphological analysis of the corpus, many such occurrences of the named entities in the test data were found given the information from the training set using orthographic and contextual features only (as they do not match exactly). However, Levenshtein distance helps to identify the variants of the same named entity. The results of applying Levenshtein distance (together with the patterns and Voting model on all features) for each category are given in Table 5 and Table 6 (model *ALL+P+L*). LOC and ORG are two categories whose performance is greatly improved by using Levenshtein distance. In case of PERSON category, recall gets slightly higher, whereas precision does not change much.

PATTERN	case
у містечку LOC in a little town LOC	locative
з містом LOC with a city LOC	instrumental
карта LOC a map of LOC	genitive
спільно з мером LOC together with a city mayor LOC	instrumental
мій рідний LOC my dear/native LOC	vocative
У LOC виявлено in LOC was found	locative
мандруючи LOC travelling in LOC	instrumental
живе десь під LOC lives somewhere nearby LOC	instrumental

Table 3: Patterns for LOC category

The last three examples in Table 4 are very interesting. They show that sometimes the nearest neighbor of the candidates for NEs in the test data is a named entity of the same category but it cannot be found by aligning. Having noticed this, we decided to exclude aligning step and to consider a nearest neighbor of every capitalized token in the test set. Although we extracted few novel person names and locations, performance in terms of precision dropped significantly. The very last example in Table 4 demonstrates a case when applying Levenshtein measure fails. In this case 'БЮТ' is of type ORG (a political party) and 'БЮТівці' are people who belong to the party. Given the nearest neighbor and the successful alignment, it is predicted that 'БЮТівці' belongs to the category ORG but it is not true. In the other example involving the same entity 'БЮТ', 'БЮТу' is correctly classified as ORG (it is the same named entity as in the training data but in dative case).

It can be concluded that, in general, Levenshtein distance helps to identify many named entities which were either misclassified or not detected at all. However, it is sometimes unable to distinguish between the variant of the same named entity and a true negative. Additional constraints such as the upper threshold of the Levenshtein distance might solve this problem.

Category	Test set	Training set	L-score
PERSON	Юлію	Юлії	1
PERSON	Лисенком	Лисенко	1
ORG	БЮТу	БЮТ	1
LOC	Львові	Львова	1
LOC	Києва	Києві	1
PERSON	Віктором	Віктор	2
PERSON	Роман	Іван	3
PERSON	Домбровський	Гошовський	4
WRONG	БЮТівці	БЮТ	4

Table 4: The nearest neighbors

As can be seen from Table 2, the best overall performance is achieved by combining contextual and orthographic features together with the patterns extracted from the Web and entities classified by employing the Levenshtein distance.

Model	PERSON	LOC	ORG
ORTHO	0.25	0.34	**0.52**
ALL	0.47	0.37	0.49
ALL+P	0.48	0.31	0.47
ALL+P+L	**0.49**	**0.55**	0.51

Table 5: Performance on each category: precision

Model	PERSON	LOC	ORG
ORTHO	**0.49**	0.26	0.42
ALL	0.36	0.15	**0.51**
ALL+P	0.36	0.27	0.49
ALL+P+L	0.42	**0.49**	0.56

Table 6: Performance on each category: recall

5 Conclusions and Future work

In this paper, we focused on standard features used for the named entity recognition on the newswire data which have been used on many languages. To improve the results that we get by employing orthographic and contextual features, we add patterns extracted from the Web and use a similarity measure to find the named entities similar to the NEs in the training set. The results we received are, in general, lower than the performance of NER systems in other languages but higher than both baselines. The former might be explained by the size of the corpus we use and by the characteristics of the language. As Ukrainian language is a language with a rich morphology, there are several directions we would like to explore in the future.

From the language-oriented perspective, it would be useful to determine to which extent stemming and morphological analysis would boost performance. The other problem which we have not considered up to now is the ambiguity of some named entities. For example, a word 'Ukraine' can belong to the category LOC as well as to the category ORG (as it is a part of a complex named entity).

In addition, we would also like to explore the semi-supervised techniques such as co-training and self-training (Collins and Singer, 1999).

References

Carreras et al. 2002. Named Entity Extraction using AdaBoost. *In the Proceedings of CoNLL-2002, Taipei, Taiwan.*

Michael Collins and Yoram Singer 1999. Unsupervised Models for Named Entity Classification. *In Proccedings of EMNLP/VLC-99.*

Radu Florian. Named Entity Recognition as a House of Cards: Classifier Stacking. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 2002.*

Dan Klein et al. Named Entity Recognition with Character-Level Models. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 2003.*

Zornitsa Kozareva, Boyan Bonev, and Andres Montoyo. 2005. Self-training and Co-training Applied to Spanish Named Entity Recognition. *In MICAI 2005: 770-779.*

Katerina Pastra, Diana Maynard, Oana Hamza, Hamish Cunningham and Yorick Wilks. 2002. How feasible is the reuse of grammars for Named Entity Recognition? *In LREC'02.*

Lance Ramshaw and Mitch Marcus. 1995. Text Chunking Using Transformation-Based Learning *In ACL'95.*

Ellen Riloff. 1995. Information Extraction as a Basis for Portable Text Classification Systems. *PhD Thesis. Dept. of Computer Science Technical Report, University of Massachusetts Amherst.*

P. P. Talukdar, T. Brants, M. Liberman and F. Pereira. 2006. A Context Pattern Induction Method for Named Entity Extraction. *In the Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-2006).*

Erik Tjong Kim Sang. 2002. Introduction to the CoNLL-2002 Shared Task: Language-Independent Named Entity Recognition. *In the Proceedings of CoNLL-2002, Taipei, Taiwan, 155–158.*

Dekai Wu et al. 2002. Boosting for Named Entity Recognition. *In the Proceedings of CoNLL-2002, Taipei, Taiwan.*

Morphological annotation of the Lithuanian corpus

Vidas Daudaravičius
Centre of Computational linguistics
Vytautas Magnus University
Donelaičio 58, Kaunas, Lithuania
vidas@donelaitis.vdu.lt

Erika Rimkutė
Centre of Computational linguistics
Vytautas Magnus University
Donelaičio 58, Kaunas, Lithuania
e.rimkute@hmf.vdu.lt

Andrius Utka
Centre of Computational linguistics
Vytautas Magnus University
Donelaičio 58, Kaunas, Lithuania
a.utka@hmf.vdu.lt

Abstract

As the development of information technologies makes progress, large morphologically annotated corpora become a necessity, as they are necessary for moving onto higher levels of language computerisation (e. g. automatic syntactic and semantic analysis, information extraction, machine translation). Research of morphological disambiguation and morphological annotation of the 100 million word Lithuanian corpus are presented in the article. Statistical methods have enabled to develop the automatic tool of morphological annotation for Lithuanian, with the disambiguation precision of 94%. Statistical data about the distribution of parts of speech, most frequent wordforms, and lemmas, in the annotated Corpus of The Contemporary Lithuanian Language is also presented.

1 Introduction

The goal of this paper is to present the experience and results of compiling a large Lithuanian morphologically annotated corpus by using an available Lithuanian morphological analyser and dealing with the disambiguation problem.

The Corpus of the Contemporary Lithuanian Language is a database of electronic texts, which is widely used in Lithuania. It well represents the present Lithuanian language and its different varieties (more about that in http://donelaitis.vdu.lt/).

```
<word="Nenuostabu" lemma="nenuostabus" type="bdvr
neig nelygin.l neįvardž bevrd.gim">
<sep=",">
<word="kad" lemma="kad" type="jngt">
<word="muziejus" lemma="muziejus" type="dktv
vyr.gim vnsk V">
<word="susilaukia" lemma="susilaukti(-ia,-ė)"
type="vksm teig sngr tiesiog.nuos esam.l vnsk IIIasm">
<word="daugelio" lemma="daugelis" type="dktv
vyr.gim vnsk K">
<word="svečių" lemma="svečias" type="dktv vyr.gim
dgsk K">
<word="ne tik" lemma="ne tik" type="jngt">
<word="iš" lemma="iš" type="prln">
<word="Čikagos" lemma="Čikaga" type="tikr dktv
mot.gim vnsk K">
<word="ir" lemma="ir" type="jngt">
<word="apylinkių" lemma="apylinkė" type="dktv
mot.gim dgsk K">
<sep=",">
<word="bet ir" lemma="bet ir" type="jngt">
<word="tolimiausių" lemma="tolimas" type="bdvr teig
aukšč.l neįvardž vyr.gim dgsk K">
<word="Amerikos" lemma="Amerika" type="tikr dktv
mot.gim vnsk K">
<word="kampelių" lemma="kampelis" type="dktv
vyr.gim dgsk K">
<word="bei" lemma="bei" type="jngt">
<word="kitų" lemma="kitas" type="įvrd mot.gim dgsk
K">
<word="šalių" lemma="šalis" type="dktv mot.gim dgsk
K">
<sep=".">
```

Figure 1: Extract from the morphologically annotated corpus (The following morphologically annotated sentence is presented: "It is no surprise that the museum is visited by guests not only from Chicago region, but also from distant American places and other countries.").

Balto-Slavonic Natural Language Processing 2007, June 29, 2007, pages 94–99,
Prague, June 2007. ©2007 Association for Computational Linguistics

Morphological annotation of the corpus will further increase capabilities of the corpus enabling extraction of unambiguous lexical and morphological information. The annotated corpus will soon be accessible for search on the internet. At the moment this corpus is fully accessible only at the Centre of Computational Linguistics of the Vytautas Magnus University. The tools for annotating Lithuanian texts are available for research purposes by request.

The Lithuanian morphological analyser *Lemuoklis* (Zinkevičius, 2000) produces results of morphological analysis of Lithuanian wordforms, but leaves unsolved the problem of morphological ambiguity. Considering successful application of statistical methods in solving the morphological ambiguity for other languages, statistical methods have also been chosen for Lithuanian. Research of morphological disambiguation and results of morphological annotation of the 100 million word Lithuanian corpus are presented in the article.

2 Morphological analysis of Lithuanian

Morphologically ambiguous wordforms are words or wordforms that have two or more possible lemma interpretations or morphological annotations, e. g. for the wordform *kovų* (en. *fights*, pl. Gen.) the morphological analyser *Lemuoklis* identifies two lemmas *kovas* (en. *rook* [bird] or *March* [month]) and *kova* (en. *fight*), while the wordform *naktis* (en. *night*) can be in Singular Nominative or in Plural Accusative case (more information on ambiguity for Lithuanian see Rimkutė, 2006).

Approximately a half of all wordforms in the Lithuanian annotated corpus are morphologically ambiguous (Rimkutė, 2006), which is comparable to other inflected languages, e.g. for the Czech language it is 46% (Hajič, 2004:173).

For developing the automatic disambiguation system a morphologically annotated training corpus is necessary. Manual creation of 1 M word Lithuanian annotated corpus is a very time consuming task, which has taken 5 man-years to complete. Firstly, the annotation format needs to be developed and mastered (see Figure 1), then it is necessary to assign a word to an appropriate part of speech, and often it is very difficult to find a correct grammatical reading for a word. It also takes a lot of time reviewing and trying to put all annotated texts into one uniform standard.

3 Automatic morphological annotation of the Lithuanian corpus

Statistical morphological disambiguation using small manually annotated training corpora looks as quite a simple task, when frequencies of grammatical features are generated during the training phase and the most likely sequence of morphological features is found in a new text by the help of various probability methods. Drawing on the experience of morphological annotation systems for other free word order languages (Dębowski, 2004; Hajič et al., 2001; Palanisamy et al., 2006 etc.), it is obvious that the corpus-based method is most suitable for the developing such systems for Lithuanian.

The Czech experience (Hladká, 2000) was very expedient for developing automatic morphological annotation tool for Lithuanian, especially because Czech similarly to Lithuanian is a free word order language. Czech research applies statistical Hidden Markov Models and formal rule-based methods for Czech and English languages. It is important to note that these methods are language independent and can be applied to Lithuanian. The only language dependent factor is a small morphologically annotated corpus for training. In various experiments the selection of Czech morphological features was regularized and optimised, which helped to achieve close to English language precision of 96%. However this precision is achieved with a limited number of Czech morphological features. The precision of 94 % is achieved when all features of Czech language are selected (Hladká, 2000).

4 Statistical morphological disambiguation

Morphologically analysed words are the input of the automatic morphological annotation system, while the best sequence of morphological features is its output. Annotation of a new text involves establishing the most likely sequence of morphological features by the help of Hidden Markov models. Not all combinations of trigrams and bigrams can be found even in the biggest corpora. Therefore, the linear smoothing of the missing cases is used, as the probability of the most likely sequence cannot be equal to zero (see more on HMM in Jurafsky (2000:305-307)).

The following HMM model is used by Czech scientists:

$$\begin{aligned}\Gamma \approx \max_T \tilde{p}(w_1 \mid t_{i_1}) * \tilde{p}(t_{i_1}) * \tilde{p}(t_{i_2} \mid t_{i_1}) * \\ * \prod_{t=3}^{n} \tilde{p}(w_t \mid t_{i_t}) * \\ * \tilde{p}(t_{i_t} \mid t_{i_{t-1}}, t_{i_{t-2}}), T = t_{i_1}, t_{i_2}, ..., t_{i_n}\end{aligned}$$

We expanded the model by including the lemma. This procedure is important to Lithuanian, where different lemmas often have identical wordforms and morphological features. Therefore the probability of a lemma is also included:

$$\begin{aligned}\Gamma \approx \max_T \tilde{p}(w_1 \mid t_{i_1}) * \tilde{p}(w_1 \mid l_{i_1}) * \tilde{p}(t_{i_1}) * \\ * \tilde{p}(t_{i_2} \mid t_{i_1}) * \prod_{t=3}^{n} \tilde{p}(w_t \mid t_{i_t}) * \tilde{p}(w_t \mid l_{i_t}) * \\ * \tilde{p}(t_{i_t} \mid t_{i_{t-1}}, t_{i_{t-2}}), T = t_{i_1}, t_{i_2}, ..., t_{i_n}\end{aligned}$$

where

$$\tilde{p}(w_t \mid t_{i_t}) = \lambda_w * p(w_t \mid t_{i_t}) + (1 - \lambda_w) * 1 / W_{t_{i_t}}$$

is the smoothed probability of a wordform and tag pair.

$$\tilde{p}(w_t \mid l_{i_t}) = \lambda_{w1} * p(w_t \mid l_{i_t}) + (1 - \lambda_{w1}) * 1 / L_{t_{i_t}}$$

is the smoothed probability of a wordform and lemma pair.

$$\tilde{p}(t_{i_t}) = \lambda_{01} * p(t_{i_t}) + (1 - \lambda_{01}) * 1 / C_T$$

is the smoothed probability of a tag.

$$\begin{aligned}\tilde{p}(t_{i_t} \mid t_{i_{t-1}}) = \lambda_{11} * p(t_{i_t} \mid t_{i_{t-1}}) + \\ + \lambda_{12} * p(t_{i_t}) + (1 - \lambda_{11} - \lambda_{12}) * 1 / C_T\end{aligned}$$

is the smoothed probability of a bigram tag .

$$\begin{aligned}\tilde{p}(t_{i_t} \mid t_{i_{t-1}}, t_{i_{t-2}}) = \lambda_{21} * p(t_{i_t} \mid t_{i_{t-1}}, t_{i_{t-2}}) + \\ + \lambda_{22} * p(t_{i_t} \mid t_{i_{t-1}}) + \lambda_{23} * p(t_{i_t}) + \\ + (1 - \lambda_{21} - \lambda_{22} - \lambda_{23}) * 1 / C_T\end{aligned}$$

is the smoothed probability of a trigram tag .

$$p(w_t \mid t_{i_t}) = \frac{Count\,(w_t \mid t_{i_t})}{Count\,(t_{i_t})}$$

is the probability of a wordform containing a particular tag in the training corpus.

$$p(t_{i_t}) = \frac{Count(t_{i_t})}{\mid T_{train} \mid}$$

is the probability of a tag in the training corpus.

$$p(t_{i_t} \mid t_{i_{t-1}}) = \frac{Count(t_{i_t}, t_{i_{t-1}})}{Count(t_{i_{t-1}})}$$

is the probability of a bigram tag in the training corpus.

$$p(t_{i_t} \mid t_{i_{t-1}}, t_{i_{t-2}}) = \frac{Count(t_{i_t}, t_{i_{t-1}}, t_{i_{t-2}})}{Count(t_{i_{t-1}}, t_{i_{t-2}})}$$

is the probability of a trigram tag in the training corpus.

$W_{t_{i_t}}$ is a number of wordforms with the feature t_{i_t}

$L_{t_{i_t}}$ is a number of lemmas with the feature t_{i_t}

C_T is a number of tags in T_{train} training set.

A function *Count(x)* corresponds to the frequency of a tag or a bigram.

Smoothing lambdas λ_{w1}, λ_w, λ_{01}, λ_{11}, λ_{12}, λ_{21}, λ_{22}, λ_{23}< 1 are used to combine the probabilities of lower order. The smoothing is very important when unknown events occur in the training corpus.

We used such lambda values:

λ_{w1}=0.85,

λ_w=0.85,

λ_{01}=0.99,

λ_{11}=0.74, λ_{12}=0.25,

λ_{21}=0.743, λ_{22}=0.203, λ_{23}=0.053

If a trigram tag is not found in the training corpus then the probability of a trigram is not assigned to zero, but rather the probability of a bigram is included with some weight. In case no trigram tag, bigram tag and unigram tag is found then the probability of a trigram assumes a very small number which is equal to 1 divided by the size of the tagset. The highest score is assigned to a trigram, lower – to bigram, and lowest – unigram.

The disambiguation tool has been developed at the Centre of Computational Linguistics of the Vytautas Magnus University using C++ tools. All results reported in this paper are based on approach using an accuracy criterion (number of correctly disambiguated results divided by number of input words). We do not use any morphological pre-processing. A precision of 94% has been achieved for establishing tags, which is comparable to results achieved for other languages, when the 1 million word training corpus is used. A precision of 99% is achieved for establishing lemmas. For the precision test a special 50 thousand word corpus has been used, which is not included in the training corpus.

The following statistics has been derived from the 1 M word training corpus[1]:

Different lemmas	*41,408*
Different pairs of wordforms and tags	*130,511*
Different pairs of wordforms and lemmas	*121,634*
Unigram tags C_T	*1,449*
Bigram tags	*76,312*
Trigram tags	*544,922*
Training corpus size $\|T_{train}\|$	*1,009,516*

Table 1: Corpus statistics

The number of lemmas in the training corpus is sufficient to gather frequencies in order to solve ambiguous lemmas. Unknown lemmas are not ambiguous in the training corpus, as they are rare and have unique meanings.

The size of the tagset is 1449. Lithuanian is a relatively free word order language, and therefore it is difficult to get reliable bigram and trigram statistics. We decided to gather distant bigram and trigram frequencies using a gap of 1. As a bigram we consider two subsequent tags (<A> <B>) or two tags with a gap of 1 in between (<A> <gap> <B>). Similarly, a trigram is a sequence of three subsequent tags (<A> <B> <C>) or a sequence of three tags with a gap of 1 between the first and second tag (<A> <gap> <B> <C>) or between the second and third tag (<A> <B> <gap> <C>). Distant n-grams help to reduce the number of unknown bigrams and trigrams in the training corpus.

5 Statistical data for the morphologically annotated corpus of Lithuanian

Most important statistical data for the morphologically annotated Lithuanian corpus:

- *Corpus size – 111,745,938 running words;*
- *Number of wordforms – 1,830,278;*
- *Number of unrecognized wordforms – 824,387 (5,6 % of all tokens);*
- *Number of recognized wordforms – 1,005,891.*

225,319 different lemmas have been recognized in the Corpus of Contemporary Lithuanian.

Distribution of parts of speech in the whole 100 M word corpus does not differ significantly from the distribution in the training corpus (see Figure 2). The biggest difference is in the number of unknown words. There are no unknown words in the training corpus, because it has been semi-automatically annotated and disambiguated. The number of unknown words in the 100 M word corpus is influenced by morphological analyzer, i.e., not all words are successfully recognized.

A big part of unknown words are proper nouns. Presently the dictionary of the morphological analyser contains 5255 high frequency proper noun lemmas (e.g. *Lietuva* (en. Lithuania)), which account for 3.2% of the vocabulary in the large annotated corpus. In the training corpus proper nouns account for 4.3% of the vocabulary, and we expect the similar proportion in the large annotated corpus. The average frequency of a proper noun lemma is 4.6 in the training corpus. Thus we could estimate the size of the dictionary of proper nouns at about 250,000 lemmas.

[1] See more about manually tagged Lithuanian Corpus and Lithuanian language tagset in Zinkevičius et al. 2005.

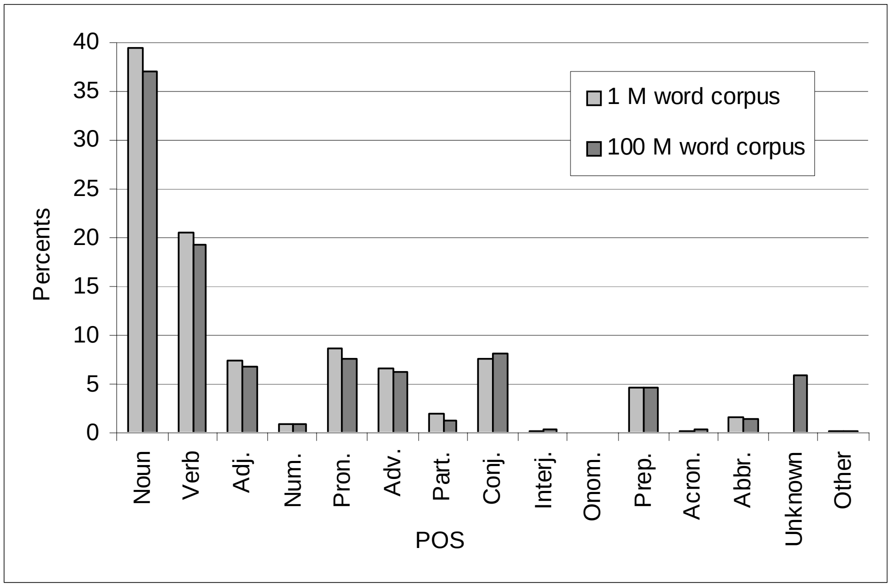


Figure 2: Distribution of parts of speech in 1 M and 100 M word corpora.

6 The remaining problems

The achieved precision of 94% for morphological annotation leaves some room for improvement. It is still difficult to solve homographic problems, where some wordforms of different words are identical. For example, wrong lemmas are frequently chosen for the wordforms *tonas* (en. *tone*) and *tona* (en. *ton*)*, kovas* (en. *rook* [bird]) and *kova* (en. *fight*)*, Biržai* (Lithuanian town) and *birža* (en. *stock-market*).

Syncretism of grammatical cases is not always solved correctly. Most often the incorrect analysis is given for words of feminine gender, when singular Genitive and plural Nominative cases are confused (e. g. *mokyklos* (en. *school*)).

Some cases are problematic even for a human linguist, when it is not clear which part of speech (noun or verb) is used in such collocations: *kovos dėl teisės likti pirmajame ešelone* (lit. *fight/ fights for the right to stay in the first league*); *kovos su narkotikais* (lit. *fight/ fights against drugs*); *kovos su okupantais* (lit. *fight/ fights against occupants*). Even if the part of speech of the word *kovos* is chosen as a noun, then the ambiguity case still remains. The broader context is needed to solve such problems.

Interjections are not very often used in Lithuanian, nevertheless the morphological abbreviation *a* is confused with the interjection *a.* Abbreviations that are identical to Roman numerals are often annotated incorrectly: the most problems are caused by the abbreviation *V.*

Sometimes wrong lemma is chosen. The words with fixed forms such as *ir* (en. *and*)*, tik* (en. *only*) cause many problems as they can be interjections, particles, or adverbs. The lemma of the wordform *vienas* (en. *one, alone, single)* is not always chosen correctly, as this word can be a pronoun, an adjective, a numeral, or even a proper noun. It is hoped that some of these problems will disappear after improving the program of morphological analysis.

7 Conclusions

The method of Hidden Markov models for morphological annotation has allowed achieving the precision of 94%, which is comparable to the precision achieved for other languages, when 1 M word training corpus is used. The precision of 99% is achieved for establishing lemmas of Lithuanian words. The precision measure estimates only the process of disambiguation, while unrecognised words are not included in the precision test.

The amount of unrecognised wordforms makes up 5,6% of all tokens (more that 800,000 different wordforms). In order to analyse the missing wordforms around 100-150 thousand lemmas need to be added to the lexicon of morphological analyser, i.e. the amount is similar to the present size of the lexicon.

One million word morphologically annotated corpus is enough for the analysis of morphological phenomena in Lithuanian, as distribution of parts of speech in the 100 million word corpus does not differ significantly

8 Acknowledgements

This work is a part of the project "Preservation of the Lithuanian Language under Conditions of Globalization: annotated corpus of the Lithuanian language (ALKA)", which was financed by the Lithuanian State Science and Study Foundation.

References:

Arulmozhi Palanisamy and Sobha Lalitha Devi. 2006. HMM based POS Tagger for a Relatively Free Word Order Language. *Research in Computing Science 18*, pp. 37-48

Barbora Vidová-Hladká. 2000. Czech language tagging. Ph.D. thesis, ÚFAL MFF UK, Prague.

Daniel Jurafsky, James H. Martin. 2000. *Speech and Language Processing*, Prentice-Hall, Upper Saddle River, NJ.

Erika Rimkutė. 2006. Morfologinio daugiareikšmiškumo ribojimas kompiuteriniame tekstyne (Morphological Disambiguation of the Corpus of Lithuanian Language). Doctoral dissertation, Vytautas Magnus University, Kaunas.

Jan Hajič. 2004. *Disambiguation of rich inflection. Computational morphology of Czech*. Karolinum Charles University, Prague.

Jan Hajič, Pavel Krbec, Pavel Květoň, Karel Oliva, Vladimír Petkevič. 2001. Serial Combination of Rules and Statistics: A Case Study in Czech Tagging. In *Proceedings of the 39Annual Meeting of the ACL (ACL-EACL 2001).* Université de Sciences Sociales, Toulouse, France.

Łukasz Dębowski. 2004. Trigram morphosyntactic tagger for Polish. In *Proceedings of the International IIS:IIPWM'04 Conference*, pp. 409-413, Zakopane.

Vytautas Zinkevičius. 2000. Lemuoklis – morfologinei analizei (A tool for morphological analysis - Lemuoklis). *Darbai ir Dienos*, 24, pp. 246–273. Vytautas Magnus University, Kaunas.

Vytautas Zinkevičius, Vidas Daudaravičius, and Erika Rimkutė. 2005. The Morphologically annotated Lithuanian Corpus. In *Proceedings of The Second Baltic Conference on Human Language Technologies*, pp. 365–370. Tallinn.

Appendix 1. Lithuanian morphological categories and appropriate tags

Grammatical Category	**Equivalent in English**	**Tag**
Abbreviation	dr.	sntrmp
Acronym	NATO	akronim
Adjective	good	bdvr
Adverb	perfectly	prvks
Onomatopoetic interjection	cock-a-doodle-do	ištk
Conjunction	and	jngt
Half participle	when speaking	psdlv
Infinitive	to be	bndr
Second Infinitive	at a run	būdn
Interjection	yahoo	jstk
Noun	a book	dktv
Number	one	sktv
Roman Number	I	rom skaič
Proper Noun	London	tikr dktv
Proper Noun2	Don	tikr dktv2
Participle	walking	dlv
Gerund	on the walk home	padlv
Preposition	on	prln
Pronoun	he	įvrd
Verb	do	vksm
Idiom AA	rest eternal	idAA
Connective idiom	et cetera	idJngt
P.S.	P.S.	idPS
Prepositional idiom	inter alia	idPrln
Pronominal idiom	nevertheless	idĮvrd
Particle	also	dll

Author Index