TermoPL — a Flexible Tool for Terminology Extraction

Małgorzata Marciniak, Agnieszka Mykowiecka, Piotr Rychlik
Institute of Computer Science PAS
ul. Jana Kazimierza 5
01-248 Warsaw, Poland
Malgorzata.Marciniak, Agnieszka.Mykowiecka, Piotr.Rychlik@ipipan.waw.pl

Abstract

The purpose of this paper is to introduce the TermoPL tool created to extract terminology from domain corpora in Polish. The program extracts noun phrases, term candidates, with the help of a simple grammar that can be adapted for user's needs. It applies the C-value method to rank term candidates being either the longest identified nominal phrases or their nested subphrases. The method operates on simplified base forms in order to unify morphological variants of terms and to recognize their contexts. We support the recognition of nested terms by word connection strength which allows us to eliminate truncated phrases from the top part of the term list. The program has an option to convert simplified forms of phrases into correct phrases in the nominal case. TermoPL accepts as input morphologically annotated and disambiguated domain texts and creates a list of terms, the top part of which comprises domain terminology. It can also compare two candidate term lists using three different coefficients showing asymmetry of term occurrences in this data.

Keywords: terminology extraction, domain corpora, Polish, C-value

1. Introduction

Every domain of language communication has its own special terminology — words and multi-word expressions. They are listed in domain glossaries and terminology dictionaries which are used both by people and computer applications. Terminology resources are useful for annotating document repositories, for better text classification and automatic translation. For newly created and rapidly changing domains, automatic terminology extraction (TE) helps in building new domain lexicons and facilitates keeping the existing ones up to dates

TE is typically performed in three steps. The first one consists in a preselection of term candidates. In the second step, the candidates are sorted according to an accepted term quality measure. Finally, general terms, i.e. terms which are used in many different domains, are filtered out by comparison of phrases obtained from domain and non-domain corpora. Various solutions for these three problems were already introduced in the relevant literature. The most interesting ones were presented in (Pazienza et al., 2005). Candidate preselection is almost always done by using simple shallow grammars which recognize the most common syntactic term structure. Term ordering is done using various strategies which can take into account: phrase frequency, number of contexts in which a phrase occurs, phrase length, and phrase elements' association measures (Pecina and Schlesinger, 2006) such as mutual information, statistical tests of independence and likelihood.

Many terminology extraction tools have already been implemented. Tools available on the Internet are adapted to process the following languages:

- English, French, Spanish, Portuguese and Italian by TermoStat (Drouin, 2003). It uses statistical and linguistic methods to identify candidate terms.
- English, Italian, French by Terminology Extraction (of Translated Labs). It uses Poisson statistics, the Maximum Likelihood Estimation and Inverse Document Frequency to rank extracted n-grams, `http://labs.translated.net/terminology-extraction/`.
- English by TerMine (Frantzi et al., 2000). It uses the C-value term candidates ranking method.

Unfortunately, not much research has been done on this subject for Polish and TermoPL is the first tool for performing the TE task which is able to take into account the structure of Polish phrases and the inflectional character of the language.

2. Terminology Extraction Issues

Automatic terminology extraction is partially a language dependent task. Methods for ranking term candidates are universal but grammar rules used for selecting terminology concepts from texts differ in various languages. The rules, which take into account morphologic analysis of text, can limit the candidates list to the syntactically correct phrases which, in turn, usually results in obtaining a better final terms list. An extraction grammar works on the results of morphologic analysis which are related to a tagset. Various languages have different tagsets. Furthermore, several tagsets for a language usually exist.

The problem of morphological variants recognition in highly inflected languages is complicated, as only nominative singular occurrences of phrases (and their homographic forms) can be directly identified by just matching them with the phrase in the base form, see the example in Table 1. Similarly, the task of recognizing terms embedded in longer phrases is more complicated than just comparing their base forms. For example, the nominative form of the phrase *kodeks prawa administacyjnego* 'code of administrative law' contains a genitive form of the nested term *prawo administracyjne* 'administrative law'. To overcome these problems we proposed in (Marciniak and Mykowiecka, 2013) to operate on simplified base forms of term candidates consisting of the base forms of their subsequent words. For the phrase in Table 1 *sklepienie kolebkowe nawy głównej* 'barrel vault of the main nave', the simplified base

form is *sklepienie*$_{subst,nom,n,sg}$ *kolebkowy*$_{adj,nom,masc,sg}$ *nawa*$_{subst,nom,f,sg}$ *główny*$_{adj,nom,masc,sg}$ ('vault' 'barrel' 'nave 'main'). It includes the shorter terms: *sklepienie kolebkowe* 'barrel vault' and *nawa główna* 'the main nave' for which the simplified base forms are, respectively, *sklepienie kolebkowy* and *nawa główny*. This approach requires the additional processing stage to convert identified terminological phrases from their simplified forms into traditional base forms to be placed on the final terminology list. This task is also language dependent, as we have to know which forms should be chosen for subsequent words. For example, in the considered phrase, the nominal genitive modifier *nawy*$_{gen,fem,sg}$ *głównej*$_{gen,fem,sg}$ 'main nave' is always in the same form in all the inflected variants of the phrase, while the adjective modifier *kolebkowe* 'barrel' is inflected together with the head noun *sklepienie* 'vault'.

The syntactic structure of nominal phrases varies for different languages. Although the nominal and adjective modifiers are very common, their ordering is not universal. For example, Polish adjective modifiers may appear before and/or after a noun which is impossible for English. To effectively extract term candidates from a particular language, it is necessary to define these rules (or to allow a user to define them) within an extraction tool. Unfortunately, none of the tools we tested allowed a user grammar to be introduced.

We decided to implement a tool dedicated to Polish but adaptable to other languages. The tool implements the C-value method proposed by (Frantzi et al., 2000), which is probably the most popular approach to TE. It was used for solving the TE task for several languages, such as English, Serbian, Slovenian, Japanese, Spanish, Chinese, Polish and Arabic. It was also compared to other methods, e.g. in a task for extracting English terminology from biomedical and general corpus (GENIA (Kim et al., 2003) and Wikipedia respectively, (Zhang et al., 2008)). In these experiments, the C-value turned out to be very good for GENIA while being much less effective for general corpus. Combined with typical keyword extraction methods like Okapi and TFIDF, the C-value was used to extract French biomedical terms from a biological laboratory test site (Lossio-Ventura et al., 2013). In (Marciniak and Mykowiecka, 2014) and (Marciniak and Mykowiecka, 2015) we discussed problems related to an application of the method to Polish and presented an evaluation of the original method and our modifications on two different corpora. Our main modification, i.e. introducing NPMI to help in identifying a nested phrase, resulted in increasing precision of the method. In this paper, we summarize these observations and describe the way they are implemented in the newly created tool, together with some new features.

3. Terminology Extraction Schema

In our extraction tool, all three phases of TE, i.e. candidate selection, ordering and filtering are now implemented. Candidate selection is done by applying a simple shallow grammar defined over lemmatized and morphologically annotated text. A default grammar is embedded in the program, but there is also possibility for introducing a grammar defined by a user, see section 5. Term ordering is performed using the slightly modified C-value coefficient defined in (Frantzi et al., 2000), which allows for comprising both one word and multi-word phrases in one terminology list. The C-value score of a phrase depends not only on the phrase frequency, but also on the number of its occurrences as a nested phrase (i.e. within other, longer ones) and the number of different phrases it occurred in. So, the method draws attention to phrases that might be important to the domain but occur within other term candidates. In this method, every phrase is assigned a C-value which is computed on the basis of the numbers of its occurrences within the text, its length and the number of different contexts it takes (within other candidate phrases). In the equation (1), for a phrase *p*, *l(p)* is a function which increases weight for longer phrases. It is equal to the logarithm of phrase length for multi-word expressions and a constant (e.g. 0.1) for one word terms. *LP* is a set of different phrases containing *p*, and *r(LP)* is the number of these phrases.

$$C\text{-}value(p) = \begin{cases} l(p) * (freq(p) - \frac{1}{r(LP)} \sum_{lp \in LP} freq(lp)) & if\ r(LP) > 0, \\ l(p) * freq(p), & if\ r(LP) = 0 \end{cases} \quad (1)$$

Frantzi et al. (2000) do not give a precise interpretation of the notion of context. This problem is discussed in (Marciniak and Mykowiecka, 2014). A number of contexts affects the C-value of a phrase, as a lower context number gives the phrase lower final score. The number of contexts may be calculated in many ways, e.g.:

- counting of pairs of left and right full contexts combined together;
- counting of pairs of left and right words combined together;
- taking into account the maximum number of different left and right word contexts counted separately.

The current version of the program counts contexts according to the second method listed above.

An important and valuable feature of the C-value method is its focusing on nested phrases. It recommends that all grammatical nested phrases recognized inside a maximal phrase candidate (longest grammar phrases in the data) should be considered as term candidates too. Unfortunately, this approach also accepts semantically odd, truncated phrases like *soft contact* as the nested phrase created from *soft contact lens*. In the paper (Marciniak and Mykowiecka, 2015), we proposed dividing a phrase according to the weakest connection between its words. In order to find that connection, we count Normalised Pointwise Mutual Information (NPMI) proposed by (Bouma, 2009) for all bigrams in a considered corpus.

The definition of this measure for the 'x y' bigram, where 'x' and 'y' are lemmas of sequence tokens, is given in Equation 2, where p(x,y) is a probability of the 'x y' bigram in the considered corpus, and p(x), p(y) are probabilities of 'x' and 'y' unigrams respectively.

Case	Singular	Plural
nom	*sklepienie kolebkowe nawy głównej*	*sklepienia kolebkowe nawy głównej*
gen	*sklepienia kolebkowego nawy głównej*	*sklepień kolebkowych nawy głównej*
dat	*sklepieniu kolebkowemu nawy głównej*	*sklepieniom kolebkowym nawy głównej*
acc	*sklepienie kolebkowe nawy głównej*	*sklepienia kolebkowe nawy głównej*
inst	*sklepieniem kolebkowym nawy głównej*	*sklepieniami kolebkowymi nawy głównej*
loc	*sklepieniu kolebkowym nawy głównej*	*sklepieniach kolebkowych nawy głównej*

Table 1: Declination of *sklepienie kolebkowe nawy głównej* 'barrel vault of the main nave'

$$NPMI(x,y) = \left(ln \ \frac{p(x,y)}{p(x)p(y)} \right) \Big/ - ln \ p(x,y) \quad (2)$$

The lowest NPMI within a phrase indicates the weakest connection within it, which suggests the best place for dividing the phrase into two parts. This process is done recursively up to unbreakable one-word phrases.

Let us consider the nested phrase creation with and without the NPMI modification in the example *nominalna roczna stopa procentowa* 'nominal annual interest rate' (from the plWikiEcono corpus (Kobyliński, 2011)). The word for word translation of the phrase is 'nominal annual rate interest'. The NPMI values of the bigrams recognized in the phrase are given in Table 2. Table 3 shows a comparison of the nested phrases obtained by the method recognizing all grammatical phrases and the NPMI modification. It may be noted that the NPMI modification, correctly, eliminated two semantically odd nested phrases from the five obtained by the first method as the strong connection of the bigram *stopa procentowa* 'rate interest' prevents it from being divided.

In the program, we implemented three slightly different versions for using NPMI measure to select nested phrases:

- The first method always divides a phrase at the weakest connection point. This method does not care if these parts satisfy grammar rules, although only grammatical phrases are finally accepted.
- The second method tries to divide phrases into subphrases so that at least one of them satisfies the grammar rules. It chooses the weakest possible connection point to make the split according to the NPMI value.
- The third method tries to divide phrases into subphrases so that at least one of them satisfies the grammar rules, but prefers cases where both subphrases obtained after splitting are accepted by the grammar. The preference can be expressed by a factor defined by the user.

For the above three methods using the NPMI value, the whole procedure is applied to all resulting subphrases recursively. The user can select one of the NPMI driven selection methods or use just the plain C-value method. Using the NPMI modification obviates supporting phrases like *giełda papierów* created from *giełda papierów wartościowych* 'stock exchange' or *spółka prawa* created from *spółka prawa handlowego* 'commercial law company' or *spółka prawa cywilnego* 'civil law partnership'. For more examples see (Marciniak and Mykowiecka, 2015); in that paper we also included an evaluation of the method on three corpora, two in Polish and one in English – GENIA. The method improved the precision for the top 1000 term candidates by 2% to 6% depending on the corpus. The method not only allowed odd terms to be removed from the top of the list, but also the number of term candidates to be lowered. In the 1.4M token set of texts concerning history of art (Art-HS), the original C-value method led to recognition of 195,623 different term candidates, while after modification we only got 154,348. More than 40K (21%) term candidates were discarded by the modified method. They were originally recognized as nested phrases only — otherwise they would be recognized by both methods. But not all such phrases were eliminated. Some of them are good term candidates and it was a motivation for introducing the C-value method. After introducing our modification, most of these particular terms are recognized on the basis of their multi variant contexts (hence, low NPMI value between the adjacent phrase elements). For example, *nowoczesne społeczeństwo* 'modern society', which occurred in these texts 6 times in 5 different contexts but never as an isolated phrase is still on the term list. On the other hand, the phrase *przykład sztuki* 'art example' which also occurred 6 times in 5 different contexts and never as an isolated phrase, was correctly not recognized as a term candidate at all.

In the term filtering scenario, the program is able to compare lists calculated for two corpora using one of the selected values: Corpora-Comparing Log-Likelihood (LL) (Rayson and Garside, 2000), Term Frequency Inverse Term Frequency (TFITF) (Bonin et al., 2010), and Contrastive Selection of Multi-Word Terms (CSmw) (Basili et al., 2001). The LL coefficient is symmetrical and shows to what extent the term distribution in two corpora is not uniform, while the next two coefficients are to be counted separately for the corpora to be compared. The higher the value, the higher indication that the term belongs to a given domain. Assuming that S_1 and S_2 denote sizes of compared corpora, f_1 and f_2 correspond to frequencies of a given term in these corpora, and $E_i = S_i \frac{f_1+f_2}{S_1+S_2}$, these values are calculated in the following way:

$$LL = 2\big(f_1 \log(\frac{f_1}{E_1}) + f_2 \log(\frac{f_2}{E_2})\big);$$

$$TFITF = log(f_1) * log\frac{S_2}{f_2};$$

Fragment	Bigram	Translation	NPMI
nominalna roczna	*nominalny roczny*	'nominal annual'	0.436
roczna stopa	*roczny stopa*	'annual rate'	0.456
stopa procentowa	*stopa procentowy*	'rate interest'	0.802

Table 2: The NPMI value for the bigrams of the phrase *nominalna roczna stopa procentowa* 'nominal annual interest rate'

The grammatically correct subphrases				NPMI driven subphrases			
'nominal'	'annual'	'rate'	'interest'	'nominal'	'annual'	'rate'	'interest'
'*nominalny*'	'*roczny*'	'*stopa*'	'*procentowy*'	'*nominalny*	'*roczny*	'*stopa*	'*procentowy*
nominalna	*roczna*	*stopa*	*procentowa*	*nominalna*	*roczna*	*stopa*	*procentowa*
nominalna	*roczna*	*stopa*		—			
	roczna	*stopa*		—			
		stopa				*stopa*	
	roczna	*stopa*	*procentowa*		*roczna*	*stopa*	*procentowa*
		stopa	*procentowa*			*stopa*	*procentowa*

Table 3: The results of two methods of nested phrases recognition for *nominalna roczna stopa procentowa* 'nominal annual interest rate'

$$CSmw = log\big(log(f_1) * \frac{f_1}{f_2/S_2}\big).$$

In the original form, these measures take into account phrase frequencies. As we wanted to rely on the ordering introduced by computing the C-value in judging term importance, we decided to use in these equations "corrected term frequencies" in these equations i.e. their C-value. The analysis of the results shows that the difference lies mainly in a lower LL (while using the C-value) for some phrases whose distribution is not very different. Such phrases are frequently judged, then, as belonging to both domains (or being general), e.g. *środek wyrazu* 'means of expression' or *poczucie humoru* 'sense of humor' (from the comparison of Art-HS and Music corpora). In TermoPL, a user may decide what he/she understands for the frequency f_i or the size S_i of a corpus. The program allows f_i to be treated as the total number of occurrences of a term in a corpus, or as its C-value. Similarly, in the equations above, S_i may stand for the sum of all occurrences of all terms, or it can be the sum of all C-values. In Table 4 an evaluation of the most contrastive terms in two relatively small corpora of about 2mln tokens is shown. The table concerns multiword terms common to both corpora with a high LL value. Terms which are relatively more frequent in the first corpus are usually domain terms while terms which are more frequent in the contrastive corpus are more often general ones. The LL coefficient is counted either using frequencies (*frq* columns) and C-value (*cv* columns).

	all		Art-HS		Music	
	cv	frq	cv	frq	cv	frq
LL>5	95	98	38	53	57	45
manual evaluation						
Art-HS domain	33	45	33	45	0	0
Music domain	14	14	0	0	14	14
rest	48	39	5	8	43	31
3<LL<=5	46	62	19	40	27	22
manual evaluation						
Art-HS domain	9	17	9	17	0	0
Music domain	5	3	0	0	5	3
rest	32	42	10	23	22	19

Table 4: Results of comparison of two corpora containing texts from different art domains: history of art (Art-HS) and music (Music) using LL

4. Program TermoPL

TermoPL is a tool that supports the process of TE from a corpus of texts concerning a domain of interest. It searches a given set of texts and creates a list of forms that might be considered as candidates for terms characteristic for a chosen domain. The program assumes that the whole set of documents was first processed by a tagger. It accepts an UTF8 encoded input with morphosyntactic analysis in three different formats: NKJP (Przepiórkowski et al., 2012, TEI), XCES, and the simple format, in which each token is represented by a single line of text consisting of an orthographic form (as it appears in a processed document), its lemma and a tag.

TermoPL reads input sentence by sentence and identifies the maximal sequences of consecutive tokens that are recognized, either by the standard built-in grammar presented in Figure 1, or a custom grammar provided by the user. In the built-in grammar, NAP and NAP_GEN both denote noun phrases, with the proviso that NAP_GEN denotes noun phrases in the genitive case. It is assumed, of course, that tokens matched by NAP (and NAP_GEN) must agree in number, case and gender. In other words, the program first extracts the longest (maximal) phrases consisting of a noun phrase, possibly modified by other noun phrases in the genitive case. Then, it splits them into smaller parts (nested phrases) that still conform to the given grammar. It provides four methods for splitting maximal phrases. The first one searches for all subphrases that satisfy the given grammar. This method produces consider-

$$
\begin{aligned}
NPP &: \$NAP\ \ NAP\_GEN^{*};\\
NAP[agreement] &: AP^{*}\ \ N\ \ AP^{*};\\
NAP\_GEN[case = gen] &: NAP;\\
AP &: ADJ \mid ADJA\ \ DASH\ \ ADJ \mid PPAS;\\
N[pos = subst, ger];&\\
ADJ[pos = adj];&\\
ADJA[pos = adja];&\\
PPAS[pos = ppas];&\\
DASH[form = \text{"-"}];&
\end{aligned}
$$

Figure 1: The built-in grammar.

ably more term candidates than the remaining three methods, since it does not care if the resulting terms are semantically odd, truncated phrases. The remaining three methods that use NPMI value for splitting phrases were described in chapter 3.

All sequences recognized in this way are converted into simplified forms, in which all words are lemmatized and stored in a set representing term candidates. Simplified forms enable the program to recognize all morphological forms of a phrase as corresponding to one term. Morphological forms of phrases may significantly differ for languages with reach inflection such as Polish. For example, *katedra romańska* 'romanesque cathedral' whose simplified form is *katedra romański* has 14 forms (e.g. *katedrze romańskiej*$_{loc,sg}$, *katedrom romańskim*$_{dat,pl}$) depending on the case and number. Two of these forms are homomorphic with the other ones.

The number of considered term candidates can be reduced by the user, if he/she submits a list of lemmas of stop words. If a term candidate contains any of the stop words, it is eliminated. For example, *ta katedra romańska* 'this romanesque cathedral' should be excluded from the list of term candidates for obvious reasons, although it conforms to the grammar used by the program. Similar problems produce compound prepositions. For example, the compound preposition *z naszego punktu widzenia* 'from our point of view' contains the grammatically valid term candidate *nasz punkt widzenia* 'our point of view', which should not be considered as a term.

Two lists are associated with each element of the set — the optional one containing all different orthographic forms of the term, and the other containing all distinct contexts in which these forms appear. The second list is automatically deleted after C-values are calculated. The first list, although it is optional, may play an important role when the base forms of terms are generated.

Additionally, for each term, two values are computed: the total number of term occurrences in the corpus and the number of occurrences within other, longer terms. Having all this information, the program calculates the C-value for each term and sorts the list of term candidates from the highest to the lowest C-value. Finally, if the user wishes to do so, simplified forms are replaced by base forms of the terms.

To obtain base forms a token or a group of tokens matched with a symbol marked with the $ character are replaced by their nominal forms. All other tokens are left unmodified. In the grammar given above, the only symbol marked with $ is NAP. Therefore all NAP phrases are transformed into their nominal forms, whereas NAP_GEN phrases are left as they appear.

In this process, the new version of Morfeusz (Woliński, 2014, the morphosyntactic analyzer and generator for Polish) is used. A base form of a term is usually singular, unless all phrases (maximal or nested) corresponding to this term are plural noun phrases. Letter case used in base forms is determined by orthographic forms associated with each term. If a particular word appears in upper case in all phrases, it remains in upper case in the base form. Otherwise, it is converted to lower case. In a case where the user decided not to collect all orthographic forms, the process of converting simplified forms to base forms relies solely on Morfeusz.

A generated list can be truncated by the user to include only multi-word terms and/or some specified number of top ranked term candidates.

The results of term extraction can be saved to a file, which in turn may be used to make comparisons with other corpora. The program calculates a selected measure for corpora similarity and marks out listed terms with different shades of colors according to their representativeness in analyzed corpora. All shades of yellow correspond to the corpus that is currently analyzed, whereas all shades of green correspond to the contrastive corpus. If the color is more saturated, it means that the frequency difference of a term is more significant for the corpus it corresponds to.

The program can be used in two modes: batch and interactive. For the interactive mode a graphical user interface is provided. The graphical user interface layout is shown in Figure 2 where the candidate list is presented. Figure 3 shows the result of two corpora comparison. TermoPL is written in Java and requires the Java Runtime Environment version 7 or later. It will comprise part of the Polish CLARIN infrastructure (`clarin-pl.eu/pl/uslugi`).

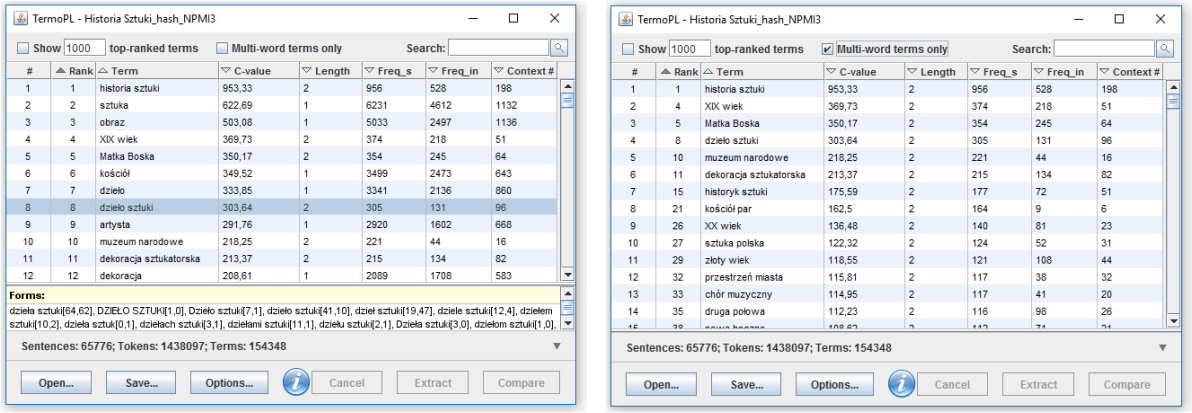

Figure 2: Exemplary results obtained from the Art-HS corpus. For each phrase, the following information is given: its position on the list, rank, base form, C-value, length, number of occurrences, number of occurrences within the context of another term and the number of these contexts. Phrases with the same C-value are ranked at the same position (second column). In the lower window, all forms of the selected term are listed together with the number of their occurrences, both in isolation and in the context of other terms. The right window contains multi word terms only (with the original ranks.)

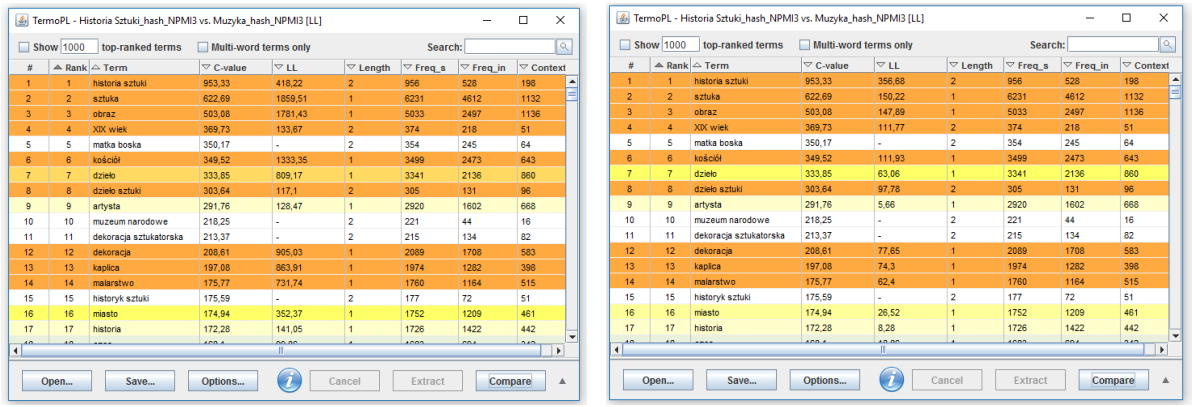

Figure 3: Results of comparing the list from Figure 2 with a list of terms obtained from a set of text about 1M tokens concerning music (mainly jazz). An additional column with the LL value is visible. Colors (gray-levels) show terms which occur in both corpora. The darker the color, the bigger the distributional difference. In the left window, comparison is made based on term frequencies; in the right window, C-values were used.

5. Customizing the Grammar

As previously mentioned, the built-in grammar can be replaced by some user-defined grammar. To specify a grammar, one has to define production rules and tests that have to be performed on tokens or sequences of tokens during the matching process. The left-hand side of a rule consists of only one nonterminal symbol. The right-hand side is a regular expression over the set of symbols. Regular expressions allowed by the program may contain alternatives separated by 'l', and quantifiers: '?', '*' and '+', which indicate zero or one, zero or more and one or more occurrences of the preceding symbol, respectively. No loops are allowed, which means that the rewriting process cannot yield to a symbol that appeared on the left hand-side of an applied rule.

For each symbol, it is possible to specify a test or a series of tests performed during the matching process. Tests can be defined on the left-hand side of a rule or in separate statements. Tests, separated with semicolons, are placed in square brackets just after a symbol to which they relate.

A test is a boolean function or an expression returning boolean value

$$\langle selector\rangle\ \langle op\rangle\ \langle string\rangle[,\ \langle string\rangle],$$

where *selector* is a function defined on tokens and lists of tokens and returning a string value, and *op* is one of the following operators: '=', '! =', '∼' and '! ∼'. The first two operators serve to compare strings if they are equal ('=') or not ('! ='). With the remaining operators, we can check whether a string returned by a selector matches ('∼') or not ('! ∼') a Java-style regular expression. If there are more strings on the right side of a positive operator ('=' or '∼'), a test succeeds whenever it succeeds for at least one of these strings. In the case of negative operators ('! =' or '! ∼'), a test succeeds if it succeeds for all given strings.

Tests can be applied to single tokens or sequences of tokens. In the simple grammar given above, $N[pos = subst, ger]$ means that a token matched with the symbol N must be a

substantive or a gerund, whereas $NAP[agreement]$ means that a sequence of tokens matched with NAP must agree in number, case and gender. Note however, that a sequence of tokens matched with NAP may contain tokens for which the $agreement$ test is not applicable, e.g. '-'. In such cases testing is performed only on those tokens for which it makes sense.

There is only one boolean function *agreement* defined in TermoPL and seven selectors whose names are self-explanatory: *form*, *lemma*, *tag*, *pos*, *number*, *case* and *gender*. This set of methods can be fairly easily augmented and modified.

6. Current and Future Extensions

Below, we mention some possible extensions of TermoPL which are currently being implemented or are intended for focusing on in the near future.

- Selectors and functions for different tagsets can be added.
- Other methods of terms ordering can be implemented.
- The list of general language terms can be provided.

The program is available from `zil.ipipan.waw.pl/TermoPL`.

7. Bibliographical References

Basili, R., Moschitti, A., Pazienza, M. T., and Zanzotto, F. M. (2001). A contrastive approach to term extraction. *Terminologie et intelligence artificielle. Rencontres*, pages 119—128.

Bonin, F., Dell'Orletta, F., Venturi, G., and Montemagni, S. (2010). A contrastive approach to multi-word term extraction from domain corpora. In *Proceedings of the 7th International Conference on Language Resources and Evaluation, Malta*, pages 19—21.

Bouma, G. (2009). Normalized (Pointwise) Mutual Information in Collocation Extraction. In *Proceedings of the Biennial GSCL Conference 2009*, pages 31–40, Tübingen. Gesellschaft für Sprachtechnologie & Computerlinguistik.

Drouin, P. (2003). Term extraction using non-technical corpora as a point of leverage. *Terminology*, 1(9):99–115.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method. *Int. Journal on Digital Libraries*, 3:115–130.

Kim, J.-D., Otha, T., Tateisi, Y., and Tsujii, J. (2003). Genia corpus – a semantically annotated corpus of biotextmining. *Bioinformatics*, 19:180–182.

Lossio-Ventura, J. A., Jonquet, C., Roche, M., and Teisseire, M. (2013). Combining c-value and keyword extraction methods for biomedical terms extraction. In *LBM'2013: 5th International Symposium on Languages in Biology and Medicine*, pages http–lbm2013.

Marciniak, M. and Mykowiecka, A. (2013). Terminology extraction from domain texts in Polish. In R. Bembenik, et al., editors, *Intelligent Tools for Building a Scientific Information Platform*, volume 467 of *Studies in Computational Intelligence*, pages 171–185. Springer-Verlag, Berlin, Heidelberg.

Marciniak, M. and Mykowiecka, A. (2014). Terminology extraction from medical texts in Polish. *Journal of Biomedical Semantics*, 5.

Marciniak, M. and Mykowiecka, A. (2015). Nested Term Recognition Driven by Word Connection Strength. *Terminology*, 21(2):180–204.

Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. (2005). Terminology Extraction: An Analysis of Linguistic and Statistical Approaches. In S. Sirmakessis, editor, *Knowledge Mining Series: Studies in Fuzziness and Soft Computing*, pages 255–279. Springer-Verlag.

Pecina, P. and Schlesinger, P. (2006). Combining association measures for collocation extraction. In N. Calzolari, et al., editors, *Proceedings of ACL, Sydney, Australia, 17-21 July 2006*. ACL.

A. Przepiórkowski, et al., editors. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN.

Rayson, P. and Garside, R. (2000). Comparing corpora using frequency profiling. In *Proceedings of the Workshop on Comparing Corpora - Volume 9, WCC '00*, pages 1—-6.

Woliński, M. (2014). Morfeusz reloaded. In N. Calzolari, et al., editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014*, pages 1106–1111, Reykjavík, Iceland. ELRA.

Zhang, Z., Irai, J., Brewster, C., and Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In *proc. of Language and Resources Conference*.

8. Language Resource References

Kobyliński, Ł. (2011). *plWikiEcono*. `http://zil.ipipan.waw.pl/plWikiEcono`.