IPhraxtor - A linguistically informed system for extraction of term candidates

Magnus Merkel, Jody Foo, Lars Ahrenberg

Department of Computer and Information Science, Linköping University

magnus.merkel@liu.se, jody.foo@liu.se lars.ahrenberg.liu.se

ABSTRACT

In this paper a method and a flexible tool for performing monolingual term extraction is presented, based on the use of syntactic analysis where information on parts-of-speech, syntactic functions and surface syntax tags can be utilised. The standard approaches to evaluating term extraction, namely by manual evaluation of the top *n* term candidates or by comparing to a gold standard consisting of a list of terms from a specific domain can have its advantages, but in this paper we try to realise a proposal by Bernier-Colborne (2012) where extracted terms are compared to a gold standard consisting of a test corpus where terms have been annotated in context. Apart from applying this evaluation to different configurations of the tool, practical experiences from using the tool for real world situations are described.

KEYWORDS: computational terminology, term extraction, evaluation, terminological work.

1 Introduction

The task of creating domain-specific terminologies can be approached with different strategies. The traditional terminological approach within terminology work, performed by terminologists, often consisted of more or less manual term extractions from texts, in combination with concept-oriented analysis of the domain. Today, much of the tedious term extraction can be replaced by automatic approaches where a computer tool can harvest term candidates from large document collections without human intervention for later processing. The results from the term candidate extraction phase is then the starting point for further selection, filtering, categorizations before the terms can be put into a term database and used by other applications, such as writing support tools, translation tools, search engines, linguistic quality assurance systems, etc.

In this paper a background to the field of automatic term extraction is presented. The background is followed by a presentation of a flexible tool for performing monolingual term extraction, based on the use of syntactic analysis where information on syntactic function and surface syntax tags can be utilised. The results from this tool can then be used in subsequent manual or automatic phases, e.g. when building a terminology or an ontology. Standard approaches to evaluating term extraction are then described and the setup for the evaluation method used in this work is described. This section is followed by a description of the corpus used in the evaluation as well as different configurations that were tested. Results for precision and recall are presented and discussed. The final section before the conclusions contains a description of how the extraction tool has been used in real world situations in the initial stages of creating term banks for Swedish government agencies. The paper ends with conclusions and a discussion of the results.

1.1 Background

Automatic term extraction is a field within language technology that deals with “extraction of technical terms from domain-specific language corpora” (Zhang et al., 2008). All approaches within automatic term extraction involve some kind of text analysis, some method of selecting and filtering term candidates. Typically the text analysis stages requires some kind of parts-of-speech analysis and lemmatization of inflected word forms, but it could also involve more complex analysis like dependency relations between tokens in the input text.

Term exctraction may have different applications in areas such as construction of ontologies, document indexing, validation of translation memories, and even classical terminology work. All the different applications share the constructive nature of the activity, and the need to distinguish terms from non-terms, or, if we prefer domain-specific terms from general vocabulary (Justeson & Katz, 1995).

The automatic term extraction process will usually be followed by a manual, sometimes computer-aided, process of *validation*. For this reason, the outputs of a term extraction process are better referred to as *term candidates* of which some, after the validation process, may be elevated to term status.

Early work on term extraction focused on linguistic analysis and POS patterns (e.g. Bourigault, 1992; Ananiadou 1994) for identifying term patterns. Later, these methods have been complemented by using more statistical measures, especially for ranking and filtering of the results (e.g. Zhang 2008; Merkel & Foo 2007).

2 The term extraction tool (IPhraxtor)

The tool presented here is in a way a standard term extraction tool relying mainly on the linguistic data. In this experiment the Connexor Machinese Syntax software (Tapanainen & Järvinen, 1997) has been used, but in principle any linguistic analyzer for any language could be used as long as the analysed texts conform to the xml dtd used. When texts are analyzed using the Connexor Machinese Syntax framework, information on various levels are produced such as base form, morpho-syntactic description (number, definiteness, case, etc.), syntactic function and dependency relation. The output data from the text analysis could look something like the following for the sentence “8. Increasing or decreasing the viscosity of the cement slurry.”:

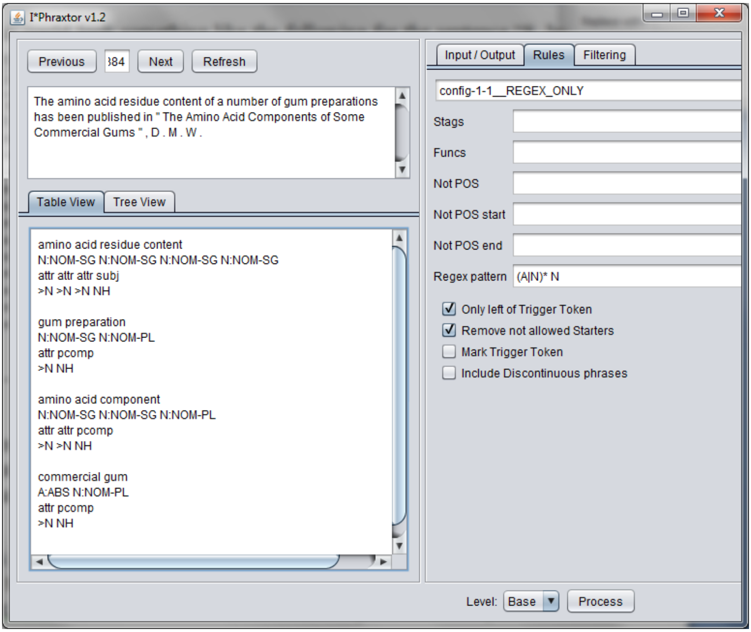
Input sentence: *8. Increasing or decreasing the viscosity of the cement slurry.*

```
XML representation after text analysis:
<s id="s25628":C04B|FILE_OR_GROUP:88303835.en.lwa|ID:38">
<w id="w637528" base="8" pos="NUM" msd="CARD" func="main" stag="NH"  >8</w>
<w id="w637529" base="." pos="INTERP" msd="Period" func="" stag="INTERP" >.</w>
<w id="w637530" base="increase" pos="V" msd="" func="" stag="VA" >Increasing</w>
<w id="w637531" base="or" pos="CC" msd="" func="cc" fa="3" stag="CC" >or</w>
<w id="w637532" base="decrease" pos="V" msd="" func="cc" fa="3" stag="VA" >decreasing</w>
<w id="w637533" base="the" pos="DET" msd="" func="attr" fa="7" stag="&gt;N" >the</w>
<w id="w637534" base="viscosity" pos="N" msd="NOM-SG" func="obj" fa="5" stag="NH" >viscosity</w>
<w id="w637535" base="of" pos="PREP" msd="" func="mod" fa="7" stag="N&lt;" >of</w>
<w id="w637536" base="the" pos="DET" msd="" func="attr" fa="10" stag="&gt;N" >the</w>
<w id="w637537" base="cement" pos="N" msd="NOM-SG" func="attr" fa="11" stag="&gt;N" >cement</w>
<w id="w637538" base="slurry" pos="N" msd="NOM-SG" func="pcomp" fa="8" stag="NH" >slurry</w>
<w id="w637539" base="." pos="INTERP" msd="Period" func="" stag="INTERP" sem="" >.</w>
</s>
```

TABLE 1. Output from the text analysis.

The information from Machinese Syntax includes the base form of each text word (“base”), morpho-syntactic features (“msd”) like number and case, dependency relations/syntactic function (“func”), syntactic surface “stag”, which can be utilised by the term extraction tool in the next step.

The analysed text is stored in xml format following the illustration in Table 1. The xml files are then loaded into the term extraction tool, called IPhraxtor. IPhraxtor is implemented in java with a graphical interface that can be used to test different strategies sentence by sentence from the input. On the Rules tab in IPhraxtor (see Figure 1), the user can specify different configurations that will govern the term extraction.



384

The amino acid residue content of a number of gum preparations has been published in " The Amino Acid Components of Some Commercial Gums " , D . M . W .

amino acid residue content
N:NOM-SG N:NOM-SG N:NOM-SG N:NOM-SG
attr attr attr subj
>N >N >N NH

gum preparation
N:NOM-SG N:NOM-PL
attr pcomp
>N NH

amino acid component
N:NOM-SG N:NOM-SG N:NOM-PL
attr attr pcomp
>N >N NH

commercial gum
A:ABS N:NOM-PL
attr pcomp
>N NH

config-1-1__REGEX_ONLY

Stags

Funcs

Not POS

Not POS start

Not POS end

Regex pattern (A|N)* N

☑ Only left of Trigger Token

☑ Remove not allowed Starters

☐ Mark Trigger Token

☐ Include Discontinuous phrases

Level: Base

Figure 1. The Rules tab in IPhraxtor where configurations of patterns and tags used can be tested.

By using the Rules tab, the user can test different regular expressions (on the POS level). In Figure 1, only the simple pattern “(A|N)* N” is used for extraction. The active sentence of the input text is shown in top left panel and below this panel the extracted term candidates, used by only using the given regular expression pattern are shown. In this case the extracted term candidates for the input sentence are “amino acid residue content”, “gum preparation” “amino acid component” and “commercial gum”.

The other fields that could be used to restrict and change the way term candidates are extracted are:

- Stags: Surface syntactic tags
- Funcs: Syntactic functions/dependency relations, such as subject, object, prepositional complement, etc.
- Not POS: POS tags that should never occur inside a term candidate
- No POS start: POS tags that are ignored at the start of term candidates
- No POS end: POS tags that are ignored at the end of term candidates
- Regex pattern: regular expression containing POS tags

Apart from giving values to these fields in order to tune the term candidate extraction process, it is also possible to specify files with stop words that will be used to trim the term candidates. The stop word lists are of two types: 1) stop words that are used at the beginning of term candidates (initial_stop_words); and 2) stopwords used at the end of term candidates (trailing_stop_words). The reason for separating the stop word lists in this fashion is that there could be cases where it is necessary to treat them differently, for example if a noun can be included in term candidate as a nominal modifier, but not as a head word. In that case, the noun in question would only be listed in the trailing stop word file, but not in the initial stop word file.

Term candidates can be exported to a database including all available information such as base forms, inflected forms, part-of-speech, frequency, and with the surrounding contexts, for later use in the terminology validation process.

The hypothesis behind extending the extraction patterns to more than just POS patterns was that some terms could actually be captured by syntactic functions (for example subjects and objects). If we use the simple regular expression used in Figure 1, term candidates like "the Save As command" will not be detected, but the candidate is indeed found if it was used as a syntactic functional subject, object or prepositional complement, and *SUBJ|OBJ|PCOMP* was specified as values in *Funcs* field.

2.1 Evaluation of terminology extraction and evaluation setup

Evaluation of term extraction systems can be done in different ways. One often used strategy is to let the system rank the output of term candidates as n-best lists where the highest ranking term candidates are at the top of the list. Then a human evaluator evaluates the top 100, 500 or 1000 term candidates with regards to whether term candidates are likely terms in the given domain. This evaluation technique has its advantages, but it also fails to capture how good the system is to capture the proportion of available terms (recall). The n-best evaluation strategy gives a good picture of how well the ranking of term candidates are and an estimate of the precision of the system, but does not provide any information on the recall of the system.

An alternative approach is to have a gold standard list of terms for a specific domain and let the output from term extraction tools be evaluated against that gold standard. This strategy was for example used in the CESART evaluation project (Mustafa el Hadi et al., 2006). In the Quæro evaluation initiative (Mondary et al., 2012), the approach of using a domain-specific terminological gold standard was further developed using the terminological precision, recall and F-measure metrics first proposed by Nazarenko et al. (2009). In principle, the output of the term extraction system is evaluated against the gold standard, thereby making it possible to compare different systems and different versions of systems applied on a specific domain. Such a gold standard should be based on the whole corpus from which terms are to be extracted. Having a gold standard also means, in essence that we already have all terms contained in a specific corpus. For practical applications, this kind of evaluation is impossible as extracting terminology from a corpus is the actual goal.

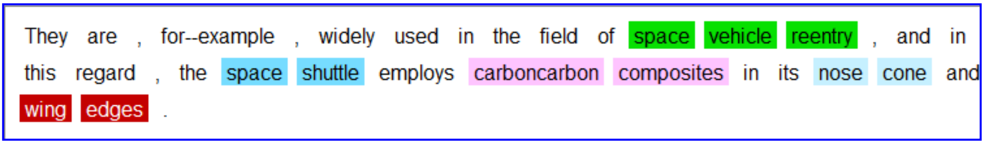
A recent proposal by Bernier-Colborne (2012) is to annotate a corpus with information on where terminological units occur in the text, "accounting for the wide variety of realizations of terms in context". Using an annotated corpus rather than a term list has several advantages. A term list will measure precision and recall on the type level, and a gold standard consisting of an annotated corpus can measure precision and recall on the token level. This is important for two reasons: (1) a terminological unit can be composed of different linguistic objects (e.g. different parts-of-speech) depending on context; and (2) linguistic tagging tools (e.g. POS taggers) are not 100 per cent accurate which can result in a decrease in either precision or recall. Having a handle on all the term candidate instances in the gold standard, makes it possible to perform error analysis and locate where the actual errors occur. Furthermore, with the right support tools, creating an annotated gold standard from a sample from the target corpus is relatively feasible for practical applications.

From a practical point of view, the Bernier-Colborne proposal, with the modification that the gold standard is created on a subset of the full corpus, is a reasonable method in order to measure the quality of a term candidate extraction system. In other words, a gold standard has to be created where terms are marked up in the subset of the actual text collection and the performance of the term candidate extraction system has to be evaluated against that gold standard. In this way, measures like precision and recall can be used in exactly the same way as they are done in for example the named entity extraction field.

The evaluation of how well different extraction strategies would work was set up in the following way:

- Creating a corpus of patent texts from five different subject areas
- Separating the corpus into a training set and a test set
- The training sets for all different texts were used to find out general strategies, using the interactive Rules tab in IPhraxtor (see Figure 1).
- Each test set consisted of 100 sentences which were marked up manually with instances of term candidates in each sentence. The aim was to only extract term candidates that were considered to be noun phrases, no attempts at catching verbs or independent adjectives were made for the evaluation. Two independent evaluators were given instructions to create the gold standards as uniformly as possible.
- Scripts for measuring precision and recall for each run against the gold standard was created.
- In the first stage ten different strategies were tested, and in the second stage, three strategies were evaluated in more detail.

The creation of the gold standard was made in a special environment where the annotator reviewed the test section of the corpus and highlighted each term candidate in the 100 sentences:



They are , for--example , widely used in the field of space vehicle reentry , and in this regard , the space shuttle employs carboncarbon composites in its nose cone and wing edges .

FIGURE 2. Simple annotation of instances of terms in the gold standard. Five terms are marked up as being terms.

The corpus used for the tests consisted of five patent application domains:

Corpus	Name	#words: en
Edible oils or fats	A23D	705,764
Cocoa	A23G	610,893
Hats: head coverings	A42B	73,991
Non-metallic elements	C01B	1,847,571
Lime, Magnesia, Slag, Cements	C04B	2,075,622

TABLE 2. Corpus description of the five different patent domains.

Initially 10 configurations were tested ranging from using one simple regular only to various setups where the use of combinations of stop words, no_start_POS, no_end_POS, Funcs, STAG was evaluated, as well as using more complex regular expressions. After this stage, it was clear that the regular expressions had contributed substantially to the overall result. Extending the simplest regular expression to other NP regular expressions only had minor effects. However, there was a clear improvement for the end result if we combined the use of stop word lists, filters for start and end POS tags, as well as functional and syntactic surface tags with the regular expression.

So if we compare the A, B and C strategies below (A using only a simple regular expression of POS tags, B using stop word lists, syntactic functional and surface tags but no regular expressions; and C using regular expression plus stopword lists)

A. Regex: (A|N)* N

B. No Regex
Func: SUBJ|OBJ|PCOMP
No_Start_POS: DET|NUM|CC|CS|PRON|PREP|ADV|INTERP
No_End_POS: DET|NUM|CC|CS|PRON|PREP|ADV|INTERP|V|A
Func: SUBJ|OBJ|PCOMP
Stag: NH|>N|N<
Stopword lists: Yes

C. Regex: "(A|N)* N"
Stopword lists: Yes

we get the following precision and recall figures for the different subcorpora[1]:

Corpus	A23D		A23G		A42B		C01B		C014B	
	P	R	P	R	P	R	P	R	P	R
RegExp (A)	0.52	0.80	0.64	0.81	0.62	0.8	0.57	0,74	0,60	0,81
Synt. Tags (no RegExp (B)	0.36	0.78	0.4	**0.91**	0.44	0.69	0.35	0.74	0.37	0.77
RegExp + stop word lists (C)	**0.63**	**0.84**	**0.72**	0.84	**0.68**	**0.82**	**0.6**	**0,76**	**0,67**	**0,82**

TABLE 3. Precision (P) and recall (R) figures for the different strategies and subcorpora**.**

In all cases we get clear increases in both recall and precision when the C strategy is used, except in A23G, where recall is actually higher for strategy B, but then with a substantially lower precision score.

[1] The scores are calculated as an average score for the two annotators of the gold standard.

We also tested several more elaborate regular expressions for catching more complex noun phrases, including coordinated noun phrases, but this did not result in better scores. Recall was slightly increased but always at the cost of lower precision. The one single factor that contributes most is whether stop words are used or not. The use of syntactic functions and syntactic surface tags does indeed lower recall in all except one of the corpora (A23G) and syntactic functions result in lower precision. Parts-of-speech tagging is usually of a very high quality in syntactic analyzers, and it could be expected that text analysers do not perform on the same level as POS tagging when it comes to detecting dependency relations and syntactic functions. This could be a factor that contributes to the poor precision rates of the B strategy, but we have no in-depth analysis to support such a claim.

3 Practical use

The term candidate extraction tool presented above has also been applied in more real life situations for making an inventory of terminology used at government agencies in Sweden. As a starting point for creating a terminology bank for Försäkringskassan (The Swedish Social Security Agency), IPhraxtor was used to explore what terminology was actually used on their external web site[2]. The idea was to assist government agency in getting a picture of what their terminology usage and combine this with a comparison of the existing term lists and other resources available on the internet, such as the Swedish National Termbank, Rikstermbanken, created by Terminologicentrum (2012).

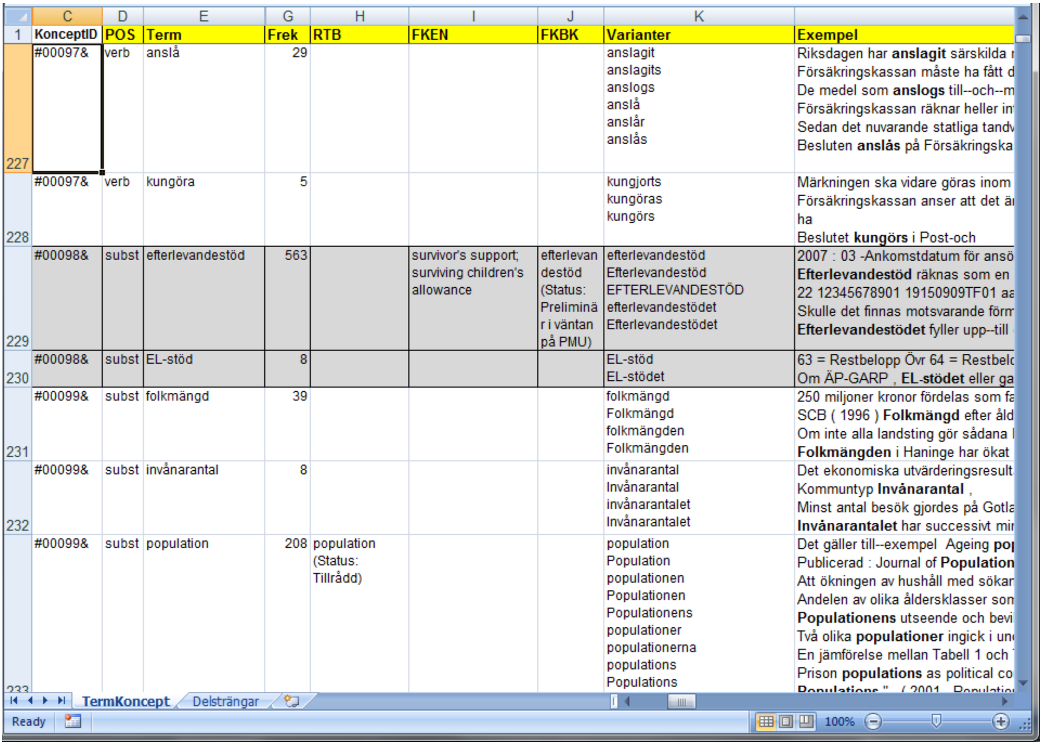
All documents from the website was processed into text format and analyzed by the Machinese Syntax tool described earlier. Then various tests were performed using the IPhraxtor tool for extracting terminology. Because all the texts were in Swedish and the expected terminology should cover verb and adjective constructions as well as noun phrases, modifications to the strategies described above were made. The terminology that existed in old term lists from the agency gave indications on what type of constructions they already used. This led to the decision that we had to prioritize recall over precision, in order to be sure that no important term candidates were missed. In Swedish it seems to be more common than in English to have prepositional complements inside noun phrase terms, for example the term “barn med funktionsnedsättning” (eng. children with disabilities). This meant that the use of the syntactic tags inside IPhraxtor could be used better as all instances of terms of this form that functioned as subjects and objects could be located, for example. The chosen strategy led to a loss of precision, as expected, but by using different ways of filtering term candidates during the validation phase, it did not cause extensive extra hours of manual work.

All in all, around 55,000 unique term candidates were extracted with the aid of IPhraxtor from the 8 million words that formed the input data from the external web site. These term candidates were then processed with a mixture of automatic, interactive and manual techniques into a remaining set of around 17.000 term candidates. The filtering process removed misspellings, term candidates in other languages than Swedish, non-words due to poor processing of PDF files as well as term candidates that simply should not be considered to be terms later on (general words that had not been filtered out by stop word lists, words that had been tagged incorrectly by the tagger, etc.). After that the 17.000 term candidates were processed in order to see if they occurred in the Swedish National Termbank (Rikstermbanken) or in any other term resource that

[2] http://www.forsakringskassan.se

already existed at the agency (e.g. Swedish English word lists). The output from this stage was handed over to Terminologicentrum and the agency as an Excel file with the intention to further categorize the term candidates into different categories and decide which term candidates that should be evaluated to “real terms” in the domain.

Examples of the resulting term candidates are shown in Figure 2:



	C	D	E	G	H	I	J	K	
1	KonceptID	POS	Term	Frek	RTB	FKEN	FKBK	Varianter	Exempel
227	#00097&	verb	anslå	29				anslagit anslagits anslogs anslå anslår anslås	Riksdagen har **anslagit** särskilda r Försäkringskassan måste ha fått d De medel som **anslogs** till--och--m Försäkringskassan räknar heller in Sedan det nuvarande statliga tandv Besluten **anslås** på Försäkringska
228	#00097&	verb	kungöra	5				kungjorts kungöras kungörs	Märkningen ska vidare göras inom Försäkringskassan anser att det är ha Beslutet **kungörs** i Post-och
229	#00098&	subst	efterlevandestöd	563		survivor's support; surviving children's allowance	efterlevandestöd (Status: Preliminär i väntan på PMU)	efterlevandestöd Efterlevandestöd EFTERLEVANDESTÖD efterlevandestödet Efterlevandestödet	2007 : 03 -Ankomstdatum för ansö **Efterlevandestöd** räknas som en 22 12345678901 19150909TF01 aa Skulle det finnas motsvarande förm **Efterlevandestödet** fyller upp--till
230	#00098&	subst	EL-stöd	8				EL-stöd EL-stödet	63 = Restbelopp Övr 64 = Restbelo Om ÄP-GARP , **EL-stödet** eller ga
231	#00099&	subst	folkmängd	39				folkmängd Folkmängd folkmängden Folkmängden	250 miljoner kronor fördelas som fa SCB (1996) **Folkmängd** efter åld Om inte alla landsting gör sådana l **Folkmängden** i Haninge har ökat
232	#00099&	subst	invånarantal	8				invånarantal Invånarantal invånarantalet Invånarantalet	Det ekonomiska utvärderingsresult Kommuntyp **Invånarantal** , Minst antal besök gjordes på Gotla **Invånarantalet** har successivt mi
233	#00099&	subst	population	208	population (Status: Tillrådd)			population Population populationen Populationen Populationens populationer populationerna populations Populations	Det gäller till--exempel Ageing **po** Publicerad : Journal of **Population** Att ökningen av hushåll med sökan Andelen av olika åldersklasser som **Populationens** utseende och bevi Två olika **populationer** ingick i und En jämförelse mellan Tabell 1 och Prison **populations** as political co **Populations** " (2001 . Populatio

FIGURE 3. Term candidates from extraction using IPhraxtor with conceptual groupings as well as information on use of term in other resources, inflectional variants and contexts where the term candidate is used.

The term candidates shown in Figure 3 are grouped as possible synonyms using the same concept ID (KonceptID), which is indicated by the fact that the term candidates “efterlevandestöd” and “EL-stöd” share the concept ID. There are also information about whether the term candidate occurs in any other resource (“efterlevandestöd” has the English translations “survivor’s support” and “surviving children’s allowance” in column FKEN (Försäkringskassan’s English term list). Also the term candidate “population” has also been located somewhere in the Swedish national term bank (Rikstermbanken) in column H, RTB.

Another place where IPhraxtor has been used is at the Swedish Unemployment Insurance Board (Inspektionen för arbetslöshetsförsäkringen – IAF). In the same way as with Försäkringskassan, the external web site was used as input for the term extraction process. Due to a smaller volume of text, the extraction stage resulted in around 12,000 term candidates which were reduced to around 3,500 term candidates during the validation stage.

The top forty “surviving” term candidates for IAF were the following:

Term candidate	**Freq.**
arbetslöshetskassa	5338
IAF	4178
arbetsförmedling	2354
arbete	1740
ärende	1526
arbetslöshetsförsäkring	1403
underrättelse	1323
ersättning	1264
sökande	1100
arbetslöshetsersättning	1086
genomströmningstid	1050
uppgift	901
arbetssökande	839
beslut	805
uppdrag	760
myndighet	668
verksamhet	625
åtgärd	542
lag	508
söka	451
kassa	440
medlem	431
intyg	414
ALF	412
förordning	411
regelverk	405
arbetslös	399
arbetsförmedlare	387
arbetsgivare	383
handläggning	381
föreskrift	370
Inspektionen för arbetslöshetsförsäkringen	369
ersättningstagare	367
utbetalning	366
medarbetare	353
aktör	352
uppföljning	344
rutin	319
arbetsgivarintyg	318
arbeta	309

TABLE 4. The top 40 term candidates from IAF

The term candidates extracted for both the government agencies have turned out to be a vital step towards creating standardized term banks. Not only did the extraction provide an insight to how terminology actually is used on their web site, it also highlighted problems regarding inconsistent usage of synonyms and acronyms as well as sets of spelling variants of terms. It was also a showcase for claiming that IPhraxtor also works for Swedish.

4 Summary and conclusion

In this paper we have presented a tool for extracting term candidates based on linguistically analyzed data, including POS, syntactic function, and surface syntax tags. The tool provides an interactive environment where combinations of regular expressions for POS sequences and the use of various linguistic tags and stop lists can be expressed and tested in a very flexible manner. We have also investigated the use of an evaluation method where a test corpus is annotated with instances of term candidates, i.e., terms in real context. This is an alternative to using terminological gold standards consisting of lists of terms for a specific domain, and, we argue that the presented approach gives another view of the quality of term extraction tools. Measuring recall for term extraction has always been a bit complicated. Even when system results are compared to a gold standard comprising of a list of validated terms for a specific domain, the results are not taking into account the specific features of how terms are distributed in the tested corpus. If the output of the system is designed to actually pinpoint each instance of a possible term in the test corpus, then it is straightforward to measure precision and recall in the same way as in, for example, named entity recognition. It could be argued that 100 sentences per subcorpora is too small a size for making this kind of evaluation; the actual number of term instances per gold standard varied between 300 and 500, but more elaborate testing would need to be performed in order to establish what size the gold standard need to be.

The evaluation from running the extraction tool on five different corpora from the patent text domain showed that the best strategy with the current tool is to use regular expressions of POS tags and stop word lists. Adding information from syntactic tags for grammatical functions like subject and object did lower the precision substantially. The most likely reason for this is that the automatic linguistic analysis did not provide complete information. However, in certain situations where more complex terms need to be captured, one might have to live with lower precision to actually bring all relevant term candidates into the validation stage. Another obstacle might be the quality of syntactic tags for grammatical functions; if linguistic analysers can provide data for grammatical functions that are as good as for parts-of-speech, then the results may be improved.

In experiences from practical applications of extracting terminology from web sites of Swedish government agencies, the tool was shown to work for Swedish and proved to be very flexible in that it provided a solid basis for further work on filtering, validating and standardizing terminology, i.e. moving from term candidates to validated terms.

Acknowledgements

This work has been funded by the Swedish Research Council (Vetenskapsrådet). Many thanks also go to Michael Petterstedt and Mikael Andersson for their help with work on software and scripting.

References

Ananiadou, S. 1994. A methodology for automatic term recognition. In Proceedings of the 15th conference on Computational linguistics, pages 1034–1038, Morristown, NJ, USA. Association for Computational Linguistics.

Bernier-Colborne, G. (2012). Defining a Gold Standard for the evaluation of Term Extractors. In: Proceedings of the colabTKR workshop Teminology and Knowledge Representation, Istanbul, Turkey.

Justeson, J.S. & Katz S.M. 1995. Technical terminology: some linguistic properties and an algorithm for identification in text. *Natural Language Engineering* 1:9-27. Bourigault, 1992;

Merkel, Foo. 2007. Terminology extraction and term ranking for standardizing term banks. In Proceedings of NODALIDA'07, Tartu, Estonia.

Mustafa El Hadi, W., Timimi, I.. Dabadie, M., Choukri, K., Hamon, O., Chiao, Y.-C. (2006). Terminological resources acquisition tools: Toward a user-oriented evaluation model. In Proceedings of LREC'06, pp. 945-958, Genova, Italy.

Nazarenko, A. & Zargayouna, H. (2009). Evaluating term extraction. In: Proceedings of the International Conference RANLO 2009, pp. 299-304, Borovets, Bulgaria.

Terminologicentrum (2012). Rikstermbanken. http:www.rikstermbanken.se.
Terminologicentrum, Stockholm.

Tapanainen, P., & Järvinen, T. (1997). A non-projective dependency parser. In Proceedings of the fifth conference on Applied Natural Language Processing (pp. 64–71). Washington, DC, USA. Stroudsburg, PA, USA: Association for Computational Linguistics.

Zhang, Z., Iria, J., Brewster, C. & Ciravegna, F. (2008). A comparative evaluation of term recognition algorithms. In: *Proceedings of LREC'08*, pp. 2108-2113, Marrakech, Morocco.