Proceedings of the Workshop

WORKSHOP ON EXTRACTING AND USING CONSTRUCTIONS IN NLP

Edited by:

Magnus Sahlgren and Ola Knutsson
SICS KTH

Workshop at NODALIDA 2009
Thursday, May 14, 2009, Odense, Denmark

SICS Technical Report T2009:10
ISSN 1100-3154

Workshop Programme
Thursday, May 14, 2009

09:00-09:10 *Opening remarks*

09:10-09:45 Jussi Karlgren: Constructions, patterns, and finding features more sophisticated than term occurrence in text (*Keynote*).

09:50-10:10 Sara Stymne: Definite Noun Phrases in Statistical Machine Translation into Danish.

10:10-10:30 Katja Keßelmeier, Tibor Kiss, Antje Müller, Claudia Roch, Tobias Stadtfeld and Jan Strunk: Mining for Preposition-Noun Constructions in German.

10:30-11:00 Coffee break

11:00-11:20 Krista Lagus, Oskar Kohonen and Sami Virpioja: Towards Unsupervised Learning of Constructions from Text.

11:20-11:40 Kadri Muischnek and Heete Sahkai: Using collocation-finding methods to extract constructions and to estimate their productivity.

11:40-12:00 Robert Östling and Ola Knutsson: A corpus-based tool for helping writers with Swedish collocations.

12:00-12:20 Gunnar Eriksson: K – A construction substitute.

12:20-13:00 *Discussion*

PREFACE

A construction is a recurring, or otherwise noteworthy congregation of linguistic entities. Examples include collocations (“hermetically sealed”), (idiomatic) expressions with fixed constituents (“kick the bucket”), expressions with (semi-) optional constituents (“hungry as a X”), and sequences of grammatical categories ([det][adj][noun]). As can be seen by these examples, constructions are a diverse breed, and what constitutes a linguistic construction is largely an open question.

Despite (or perhaps due to) the inherent vagues of the concept, constructions enjoy increasing interest in both theoretical linguistics and in natural language processing. A symptom of the former is the Construction grammar framework, and a symptom of the latter is the growing awareness of the impact of constructions on different kinds of information access applications. Constructions are an interesting phenomenon because they constitute a middleway in the syntax-lexicon continuum, and because they show great potential in tackling infamously difficult NLP tasks.

We encouraged submissions in all areas of constructions-based research, with special focus on:

- Theoretical discussions on the nature and place within linguistic theory of the concept of linguistic constructions.
- Methods and algorithms for identifying and extracting linguistic constructions.
- Uses and applications of linguistic constructions (information access, sentiment analysis, tools for language learning etc.).

We received 8 submissions and accepted 6 papers. The six accepted papers are published in these proceedings. Each paper was reviewed by two members of the Program Committee.

Acknowledgements
We wish to thank our Program Committee: Benjamin Bergen (University of Hawaii), Stefan Evert (University of Osnabrück), Auður Hauksdóttir (University of Iceland), Emma Sköldberg (University of Gothenburg), and Jan-Ola Östman (University of Helsinki).

Stockholm, May, 2009

Magnus Sahlgren and Ola Knutsson

Definite Noun Phrases in Statistical Machine Translation into Danish

Sara Stymne
Xerox Research Centre Europe, Grenoble, France
Linköping University, Linköping, Sweden
sarst@ida.liu.se

Abstract

There are two ways to express definiteness in Danish, which makes it problematic for statistical machine translation (SMT) from English, since the wrong realisation can be chosen. We present a part-of-speech-based method for identifying and transforming English definite NPs that would likely be expressed in a different way in Danish. The transformed English is used for training a phrase-based SMT system. This technique gives significant improvements of translation quality, of up to 22.1% relative on Bleu, compared to a baseline trained on original English, in two different domains.

1 Introduction

A problematic issue for machine translation is when a construction is expressed differently in the source and target languages. In phrase-based statistical machine translation (PBSMT, see e.g. Koehn et al. (2003)), the translation unit is the phrase, i.e., a sequence of words, which can be contiguous or non-contiguous. Short range language differences can be handled as part of bi-phrases, pairs of source and target phrases, if they have been seen in the training corpus. But structural differences cannot be generalized.

A construction that is different in Danish compared to many other languages is the definite noun phrase. In Danish there are two ways of expressing definiteness, either by a suffix on the noun or by a definite article (1). In many other languages definiteness is always expressed by the use of definite articles, as in English where *the* is used.

(1) *den rette anvendelse af fondene*
DEF right use of funds-DEF
the proper use of the funds

This difference causes problems in translation, with spurious definite articles or the wrong form of nouns, as in (2), where an extra definite article is inserted in Danish.

(2) *det skal integreres i *den traktaten*
it shall integrate in DEF treaty-DEF
it must be integrated in the treaty

In this paper we propose a method for identifying English definite NPs that are likely to be expressed by a definite suffix in Danish. These phrases are then transformed to obtain *Danish-like* English. The algorithm is based on part-of-speech tags, and performed on English monolingual data. The Danish-like English is used as training data and input to a PBSMT system.

We evaluate this strategy on two corpora, Europarl (Koehn, 2005) and a small automotive corpus, using Matrax (Simard et al., 2005), a phrase-based decoder that can use non-contiguous phrases, i.e. phrases with gaps. We investigate the interplay between allowing gaps in phrases and using preprocessing for definiteness. We find that using non-contiguous phrases in combination with definiteness preprocessing gives the best results, with relative improvements of up to 22.1% over the baseline on the Bleu metric (Papineni et al., 2002).

2 Definiteness in Danish

Definiteness in Danish can be expressed in two ways, either by a suffix on the noun (*-en/-et* etc.), or by a prenominal definite article (*den/det/de*). The definite article is used when a noun has a premodifier, such as an adjective (3) or a numeral (4). In other cases, definiteness is expressed by a suffix on the noun (5).

(3) *det mundtlige spørgsmål*
DEF oral question
the oral question

(4) *de 71 lande*
DEF 71 countries
the 71 countries

(5) *kommissionen og rådet*
commission-DEF and council-DEF
the commission and the council

The distribution of the type of definiteness marking is fixed in standard Danish; the suffix cannot be used with a pre-modifier, and the definite article cannot be used for bare nouns. Only one type of definite marking can be used at the same time, which is different to most of the other Scandinavian languages, where double definiteness occur. There are, however, some subtleties involved. For instance, Hankamer and Mikkelsen (2002) pointed out that either type of definite marking can be used for a bare noun post-modified by a relative clause, rendering either a restrictive or non-restrictive interpretation. This, however, will not be taken into account further in this study.

3 Related Work

In a PBSMT system short range transformations and reorderings can be captured in bi-phrases that contain these phenomena. These include phenomena such as adjective-noun inversion in English to Italian translation (e.g., *civil proceedings – procedura civile*), which works well for phrases that have been seen at training time. However, the system cannot generalize this knowledge. For phrases it has not already seen in the training corpus, it has to rely on the language model to favour an idiomatic target sequence of words. The language model, however, has no knowledge of the source sentence.

There have been many suggestions of hierarchical models for statistical machine translation that go beyond the power of PBSMT, and can model syntactic differences. Syntax can be used either on the source side (Liu et al., 2006), the target side (Yamada and Knight, 2002), or on both sides (Zhang et al., 2007a). These models are all parser-based, but it is also possible to induce formal syntax automatically from parallel data (Chiang, 2005). While several of these approaches have shown significant improvements over phrase-based models, their search procedures are more complex, and some methods do not scale well to large training corpora.

One way to address the issue of constructions being realised in different ways, still in the framework of PBSMT, is by preprocessing the training data to make one of the languages similar to the other, which has been applied for instance to German phrasal verbs, compounds in Germanic languages, and word order in many languages.

Nießen and Ney (2000) described work where they performed a number of transformations on the German source side for translation into English. One of the transformations was to join separated verb prefixes, such as *fahre ... los/losfahren* (*to leave*) to the verb, since these constructions are usually translated with a single verb in English.

A construction that has received a lot of attention is the compound. Compounds are normally written as one word without any word boundaries in Germanic languages, and as two words in English and many other languages. A common strategy is to split compounds into their components prior to training and translation for German (Nießen and Ney, 2000; Popović et al., 2006) and Swedish (Stymne and Holmqvist, 2008), but also the opposite, to merge English compounds, has been investigated (Popović et al., 2006).

Preprocessing is a common way to address word order differences for many language pairs. A common strategy is to apply a set of transformations to the source language prior to training and decoding. The transformations can be handwritten rules targeting known syntactic differences (Collins et al., 2005), or they can be learnt automatically (Habash, 2007). In these studies the reordering decision is taken deterministically on the source side. This decision can be delayed to decoding time by presenting several reordering options to the decoder (Zhang et al., 2007b; Niehues and Kolss, 2009). In one of the few studies on SMT for Danish, Elming (2008) integrated automatically learnt reordering rules into a PBSMT decoder. Reordering rules can be learnt using different levels of linguistic annotation, such as part-of-speech (Niehues and Kolss, 2009), chunks (Zhang et al., 2007b) or parse trees (Habash, 2007).

While there has been a lot of work on preprocessing for SMT, to the best of our knowledge, there has not been much focus on definiteness. We are only aware of one unpublished study that targets definiteness. Samuelsson (2006) investigated if SMT between Swedish and German could be improved by transforming raw German text so that

it became more similar to Swedish with regard to definiteness.

4 Preprocessing of English

Our goal is to transform English definite NPs so that they become similar in structure to Danish NPs. When definiteness is realised with a definite article in Danish, we want to preserve the English source as it is, but when it is realised by a suffix, we want to transform the English, by removing the definite article as a separate token, and add it as a suffix to the main noun. Example results of this process is shown in (6–8).

(6) *the commission*
DET NOUN
commission#the

(7) *the member states*
DET NOUN NOUN
member states#the

(8) *the old commission*
DET ADJ NOUN
the old commission

The transformations are based on part-of-speech tags, from an in-house Hidden Markov model tagger, based on Cutting et al. (1992). On the tagged output we identify definite NPs by looking for the definite article *the*. If *the* is followed by at least one noun, it normally corresponds to a suffix construction in Danish, and hence it is removed and a suffix is added to the last consecutive noun, which can either be a single noun (6), or the head of a compound noun (7). If *the* is not directly followed by a noun, we assume that it is followed by some modifier, in which case definiteness is expressed by an article in Danish, so no transformation is performed (8). In summary, we perform the following steps:

```
foreach English word/POS–tag pair:
  if word == 'the':
     if next POS–tag == 'NOUN':
       remove 'the', and add a suffix
       to the last consecutive noun
```

The identification is performed monolingually on the source side, assuming that English definite NPs have the same distribution as Danish definite NPs, which is not always the case. An alternative would have been to train a classifier based on word alignments with Danish. Another alternative would have been to identify NPs by using either a chunker or a parser. However, the fact that the distribution rules in Danish are simple and general, made us believe that simple part-of-speech-based rules was good enough for this type of identification, and could definitely show the feasibility of the main approach.

A drawback of our transformations is that we risk introducing data sparsity, by transforming English nouns into new tokens, marked for definiteness.

5 System Description

In all experiments we use the phrase-based decoder Matrax (Simard et al., 2005), developed at Xerox. Matrax is based on a fairly standard log-linear model:

$$Pr(t|s) = \frac{1}{Z_s} exp\left(\sum_{m=1}^{M} \lambda_m h_m(t,s)\right) \quad (9)$$

The posterior probability of a target sentence t given a source sentence s is estimated by M feature functions h_m, which are all assigned a weight λ_m. Z_s is a normalization constant. The following feature functions are used:

- Two bi-phrase feature functions, i.e., the probability of a sequence of source phrases based on the corresponding sequence of target phrases, and reversed: $Pr(t|s)$ and $Pr(s|t)$
- Two compositional bi-phrase feature functions, as above, but the probabilities are based on individual word translations, not on full phrases: $Lex(t|s)$ and $Lex(s|t)$
- A 3-gram language model trained by the SRILM toolkit (Stolcke, 2002) on the Danish side of the parallel corpus
- A number of penalty feature functions:
  - Word count
  - Bi-phrase count
  - Gap count
  - Distortion penalty, measuring the amount of reordering between bi-phrases in the source and the target

The weights, λ_m, of the feature functions are estimated against a development corpus by maximizing a smoothed NIST function using gradient-based optimization techniques (Simard et al., 2005).

		Automotive		Europarl	
		English	Danish	English	Danish
Training:	Sentences	168046		701157	
	Running words+punctuation	1718753	1553405	14710523	13884331
	Vocabulary	16210	31072	67434	175764
Development:	Sentences	1000		1000	
	Running words+punctuation	10100	9078	21502	20062
	Vocabulary	1991	2183	3241	3857
Test:	Sentences	1000		1000	
	Running words+punctuation	10128	9358	20396	18449
	Vocabulary	2019	2300	4532	5116

Table 1: Corpus statistics

Matrax is original in that it allows non-contiguous bi-phrases, such as *jeopardise – bringe ... i fare (bring ... into danger)*, where words in the source, target, or both sides can be separated by gaps that have to be filled by other phrases at translation time. Most other phrase-based decoders can only handle phrases that are contiguous. We also simulate a standard PBSMT decoder with only contiguous phrases, by using Matrax and filtering out all bi-phrases that contain gaps.

For the automotive corpus, we run a separate module that replaces digits and units with placeholders prior to training and translation. These are replaced after translation by the corresponding digits and units from the source. We also translate content within brackets separately, in order to avoid reordering that crosses brackets. These modules are not used in the Europarl experiments.

6 Experiments

We perform experiments on two corpora. One is a small corpus of automotive manuals, extracted from a translation memory. The other is a part of the larger and more diverse Europarl corpus (Koehn, 2005) of transcribed European parliament speeches. In the automotive corpus sentences longer than 55 words were filtered out, and in Europarl, sentences longer than 40 words. Table 1 gives details of the two corpora. Besides being larger, Europarl is also more complex, with longer sentences, and more diverse vocabulary, and can be expected to be a harder corpus for machine translation.

For both corpora we perform translation from English to Danish, applying definiteness preprocessing for English (*DP*). We compare this to a baseline without definiteness preprocessing (*Base*). We also investigate how definite preprocessing interplay with PBSMT systems with and without gaps in the bi-phrases ($+/-$*Gaps*). In the condition where gaps are allowed, we allow up to four gaps per bi-phrase.

		Bleu	NIST
+Gaps	Base	70.91	8.8816
	DP	76.35	9.3629
−Gaps	Base	73.86	9.1510
	DP	73.74	9.1504

Table 2: Translation results on the automotive corpus

Results are reported using two automatic metrics, Bleu (Papineni et al., 2002) and NIST (Doddington, 2002), calculated on lower-cased data. Statistical significance testing is performed using approximate randomization (Riezler and Maxwell, 2005), with $p < 0.01$.

6.1 Results

Table 2 shows the results for the automotive corpus. Generally the scores are very high, showing that it is an easy corpus to translate. When we use gaps, the definite preprocessing gives a relative increase of 7.7% on Bleu. When no gaps are used, the definite preprocessing does not make a significant difference to the scores. Both systems without gaps are significantly better than the baseline with gaps, but significantly worse than the combination of gaps and definiteness preprocessing.

Table 3 shows the results for Europarl. Here we have an even larger relative improvement on Bleu, of 22.1%, when adding definiteness preprocessing with gaps. In this case the definite preprocessing also gives a significant improvement, of 14.0%, when used without gaps. Again we see an improvement for the baseline without gaps over that with gaps, whereas there is no significant difference with definite preprocessing with and without gaps.

Source	The majority of the women will be travelling to a conference of members of parliament in Berlin.
Reference	Hovedparten af kvinderne skal af sted til en konference for parlamentsmedlemmer i Berlin.
DP+Gaps	Flertallet af kvinderne bliver rejser til en konference af medlemmer af parlamentet i Berlin.
DP−Gaps	Flertallet af kvinderne vil være rejser til en konference af medlemmer af parlamentet i Berlin.
Base+Gaps	Størstedelen af de kvinder bliver til en konference for parlamentsmedlemmer rejser i Berlin.
Base−Gaps	Størstedelen af de kvinder bliver rejse til parlamentsmedlemmerne i en konference i Berlin.

Figure 1: Example translations

		Bleu	NIST
+Gaps	Base	19.01	5.6373
	DP	23.22	6.1009
−Gaps	Base	20.40	5.8613
	DP	23.26	6.0308

Table 3: Translation results on Europarl

The overall scores are much lower for Europarl, as can be expected on the basis of the corpus characteristics, despite the larger amount of training data. The definiteness preprocessing is, however, useful on both corpora, particularly in combination with gaps.

The definiteness preprocessing increases the English vocabulary, by introducing new noun tokens, marked for definiteness. This does not seem to be a serious problem, however, since in the Europarl test set there were only seven such tokens that are unknown for the systems with definiteness preprocessing, of which three were also unknown in the baseline systems. In the automotive test set the problem was even smaller, with two and three such unknown tokens with and without gaps.

Figure 1 shows the translations produced by the different systems for a Europarl sentence. With regard to definiteness, the systems with definiteness preprocessing perform better, producing *kvinderne* (*the women*) with a suffix instead of a definite article. There are also different word choices, for instance for the first phrase, *the majority*, for which all options are acceptable. Another difference is that *members of parliament* is produced as the desired compound in the baseline systems, but as a complex noun phrase with definiteness preprocessing. All systems fail to produce a good translation of *will be travelling*, but the baseline system with gaps also misplaces the main verb *rejser* (*travels*), towards the end of the sentence.

As we can see in Figure 1, and in many other sentences, the definiteness construction is improved by the use of preprocessing. But the preprocessing also leads to other changes in translations, which are both positive and negative, such as different word choice and word order. We believe it would be useful to investigate such changes further by a thorough error analysis.

7 Conclusion and Future Work

By targeting one single construction with simple preprocessing, we can achieve significant improvements in translation quality, which was shown on two very different corpora, with respect to size, sentence length, and diversity. This suggests that language pair specific identification and transformation of constructions that differ between languages is a useful way to improve the quality of phrase-based statistical machine translation.

In our current system, we make a discriminative choice of which definite construction to use in the transformed English that will not always be the best choice. A way to handle this is by using lattice input to the decoder, which delays the choice of which construction to use. Yet another possibility would be to integrate the transformation rules into the decoder, in a similar manner to the reordering rules used by Elming (2008). It would also be interesting to combine definiteness preprocessing with other ways to harmonize languages, such as reordering and compound splitting.

The fact that definiteness can be expressed by a suffix holds true also for the other Scandinavian languages. However, the distribution is somewhat different, with phenomena such as double definiteness in some languages. With a few modifications to the identification and transformation rules for English, we believe that the same method is likely to be useful also for translation into other Scandinavian languages. The source language does not need to be constrained to English, but could be any language where definiteness is expressed by definite articles.

Acknowledgement

Thank you to Nicola Cancedda, Tamás Gaál, Francois Pacull, and Claude Roux at XRCE for the in-

troduction to the Xerox tools, useful discussions on the approach, and comments on different versions of this paper.

References

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 263–270, Ann Arbor, Michigan.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 531–540, Ann Arbor, Michigan.

Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. 1992. A practical part-of-speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 133–140, Trento, Italy.

George Doddington. 2002. Automatic evaluation of machine translation quality using n-gram co-occurence statistics. In *Proceedings of the Second International Conference on Human Language Technology*, pages 228–231, San Diego, California.

Jakob Elming. 2008. Syntactic reordering integrated with phrase-based SMT. In *Proceedings of the Second ACL Workshop on Syntax and Structure in Statistical Translation*, pages 46–54, Columbus, Ohio.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of MT Summit XI*, pages 215–222, Copenhagen, Denmark.

Jorge Hankamer and Line Mikkelsen. 2002. A morphological analysis of definite nouns in Danish. *Journal of Germanic Linguistics*, 14(2):137–175.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology and North American Association for Computational Linguistics Conference*, pages 48–54, Edmonton, Alberta, Canada.

Philipp Koehn. 2005. Europarl: a parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 609–616, Sydney, Australia.

Jan Niehues and Muntsin Kolss. 2009. A POS-based model for long-range reorderings in SMT. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 206–214, Athens, Greece.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 1081–1085, Saarbrücken, Germany.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318, Philadelphia, Pennsylvania.

Maja Popović, Daniel Stein, and Hermann Ney. 2006. Statistical machine translation of German compound words. In *Proceedings of FinTAL – 5th International Conference on Natural Language Processing*, pages 616–624, Turku, Finland. Springer Verlag, LNCS.

Stefan Riezler and John T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing for MT. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 57–64, Ann Arbor, Michigan.

Yvonne Samuelsson. 2006. Nouns in statistical machine translation. Unpublished manuscript (Term paper, Statistical Machine Translation), Copenhagen Business School, Denmark.

Michel Simard, Nicola Cancedda, Bruno Cavestro, Marc Dymetman, Eric Gaussier, Cyril Goutte, Kenji Yamada, Philippe Langlais, and Arne Mauser. 2005. Translating with non-contiguous phrases. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 755–762, Vancouver, British Columbia, Canada.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado.

Sara Stymne and Maria Holmqvist. 2008. Processing of Swedish compounds for phrase-based statistical machine translation. In *Proceedings of the European Machine Translation Conference*, pages 180–189, Hamburg, Germany.

Kenji Yamada and Kevin Knight. 2002. A decoder for syntax-based statistical MT. In *Proceedings of the 40th Annual meeting of the ACL*, pages 303–310, Philadelphia, Pennsylvania.

Min Zhang, Hongfei Jiang, Ai Ti Aw, Jun Sun, Sheng Li, and Chew Lim Tan. 2007a. A tree-to-tree alignment-based model for statistical machine translation. In *Proceedings of MT Summit XI*, pages 535–542, Copenhagen, Denmark.

Yuqi Zhang, Richard Zens, and Hermann Ney. 2007b. Improved chunk-level reordering for statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 21–28, Trento, Italy.

Mining for Preposition-Noun Constructions in German

Katja Keßelmeier | **Tibor Kiss** | **Antje Müller** | **Claudia Roch** | **Tobias Stadtfeld** | **Jan Strunk**

Sprachwissenschaftliches Institut
Ruhr-Universität Bochum
D-44801 Bochum, Germany

{kesselmeier, tibor, mueller, roch, stadtfeld, strunk}@linguistics.rub.de

Abstract

Preposition-noun constructions (PNCs) are problematic in that they allow the realization of singular count nouns without an accompanying determiner. While the construction is empirically productive, it defies intuitive judgments. In this paper, we describe the extraction of PNCs from large annotated corpora as a preliminary step for identifying their characteristic properties. The extraction of the data relies on automatic annotation steps and a classification of noun countability.

1 Introduction

In many languages, the realization of a determiner with a singular count noun is mandatory. Yet, the same languages show apparently exceptional behaviour in allowing the combination of prepositions with *determinerless* nominal projections. Minimally, such a construction consists of a preposition and an unadorned count noun in the singular, as illustrated in (1).

(1) *auf Anfrage* (after being asked), *auf Aufforderung* (on request), *durch Beobachtung* (through observation), *mit Vorbehalt* (with reservations), *unter Androhung* (under threat)

The construction is not restricted to the collocation-like combinations in (1). It can be extended in all possible ways allowed for nominal projections with the characteristic property that the resulting projection does not contain a determiner. More complex constructions are illustrated in (2).

(2) *auf parlamentarische Anfrage* (after being asked in parliament), *bei absolut klarer Zielsetzung* (given a clearly present aim), *mit schwer beladenem Rucksack* (with heavily loaded backpack), *nach mehrfacher Verschiebung der Öffnung* (after postponing the opening several times), *unter sanfter Androhung* (under gentle threat)

Sometimes, the constructions in (1) and (2) have been called *determinerless PPs* (cf. Quirk et al., 1985). Since determiners combine with nominal projections, and not with prepositions, we will refrain from using this terminology and call the phrases in (1) and (2) *preposition-noun constructions* (henceforth: PNCs).

Until recently, PNCs have been considered as exceptions in both theoretical and computational linguistics. A striking example is their treatment in the Duden grammar of German, which considers the realization of a determiner with a singular count noun mandatory and treats PNCs as exceptions that can be listed. But Baldwin et al. (2006) have pointed out that the equivalent construction is productive in English; Dömges et al. (2007) have verified the empirical productivity of the construction in German on the basis of a stochastic model. They remark, however, that the empirical productivity of the construction does not correspond to its intuitive productivity: while speakers of German are able to understand PNCs occurring in newspaper texts, they are reluctant

to coin new PNCs. Hence, the linguist is confronted with a phrasal combination whose properties cannot easily be determined by introspective judgments. Consequently, it still remains unclear which factors allow a singular count noun to appear without an article if embedded under a preposition.

It has also been assumed that PNCs and PPs can be distinguished by the simple fact that PNCs do not show up as ordinary PPs. That is, it should not be possible to transform a PNC into a PP by just adding a determiner. However, this assumption is not correct. In example (3), we can either use a PNC or a PP containing a singular NP or a PP containing a plural NP without any change in its grammaticality (and only the slightest changes in interpretation).

(3) *Milosevic unterschrieb*
Milosevic signed
auch unter Androhung/der
even under threat_{sg}/the
Androhung/Androhungen
threat/threats_{pl}
von NATO-Bombardementen
of NATO-air-raids
nicht.
not
'Milosevic did not even sign on pain of NATO air raids.'

As speaker intuition cannot be used to determine the properties of this construction, we are pursuing an alternative strategy. We assume that the constitutive properties of PNCs can be determined by making use of *Annotation Mining* (Chiarcos et al., 2008). To this end, we annotate large corpora both automatically and manually, and extract the pertinent constructions from the annotated corpora, including not only PNCs, but also corresponding PPs (as illustrated in (3)), in order to determine the characteristics that distinguish PNCs from ordinary PPs.

The data are extracted from a large newspaper corpus, the *Neue Zürcher Zeitung corpus* (1993-1999), which contains approximately 200 million words. Carried out as a case study, we have initially opted for an inline XML format, but will move on to a stand-off format. We use standard tools available for the analysis of large corpora for automatic annotation, in particular two part-of-speech taggers, a morphological analyser and a phrasal chunker. In addition, we had to develop a genuine classifier for noun countability, since we are only interested in those PNCs in which the noun is classified as countable. Noun countability cannot be determined as a lexical property but must be considered a contextual property (cf. Allan, 1980). To this end, we have developed a classification system by chaining together a decision tree and a naïve Bayes classifier.

In section 2, we describe the automatic morphosyntactic and categorial annotation of the corpus provided by two different taggers and present the classification of noun countability. Section 3 describes the indexing and search procedures. We also present a small-scale evaluation of the extraction method. Section 4 briefly describes manual annotation steps that are further required to carry out annotation mining.

2 Corpus processing

2.1 Construction of the corpora

The construction of the corpora started with plain-text files for each volume of the NZZ newspaper from 1993 to 1999. The first step was to identify the document structure and to extract meta-information about genre, date, and author for each article. Since headlines and titles are often formulated in telegraph style with anomalous use of articles, it was very important to determine the membership of a sentence in a title-section or a paragraph. This was done using simple heuristic methods. To further facilitate the preprocessing, the daily issues of the newspaper were stored in 2092 individual files. A daily issue contains approx. 98,000 tokens on average, which turned out to be a size that could be handled well by all tools employed.

2.2 Automatic morphosyntactic analysis

The tokenization and sentence-boundary detection of the corpora was performed using the *Punkt* system (Kiss and Strunk, 2006). After converting the data into the customary format for tagging (one token per line), two taggers were used simultaneously to process the corpora.

The *Regression-Forest Tagger* (Schmid and Laws, 2008) does not only produce POS tags but also performs a morphological analysis of each token based on SMOR (Schmid et al., 2004). It thus provides the lemma and morphosyntactic features of nouns, including their number value and whether we are dealing with a common or proper noun. To maximize the quality of the morphological analysis we trained the morphological component of the RFT on a full lexicon of all word types occurring in our corpora. The

high accuracy of this tagger for identifying the number value of nouns (a preliminary test resulted in over 97% accuracy) was the main argument for using RFT.

The *TreeTagger* (Schmid, 1995) provides POS annotations as well, but in addition determines non-recursive chunks essential for the identification of PNCs and regular PPs.

To aggregate the output of the two taggers in a standard common format, we have not only integrated the annotations of the two taggers for each daily issue into a single valid inline XML data format, but also reorganized and enhanced the previously extracted meta-information, and defined an individual ID for every token, sentence, segment, and article. The user can thus identify sentences or tokens unambiguously even in huge corpora and across different preprocessing and annotation tools. Table 1 exemplifies the token ID *NZZ_1994_04_27_a32_seg5_s13_t4* and Figure 1 shows a small example of the constructed inline XML data format.

Name of newspaper	NZZ
Year	1994
Month	04
Day	27
Number of article (in daily issue)	32
Number of segment (in article)	5
Number of sentence (in segment)	13
Number of token (in sentence)	4

Table 1. Structure of the global IDs.

2.3 Countability classification

Allan (1980) suggests that countability is not a lexical property, but determined by the formal context of a noun. Nevertheless, his classification system accounts for the fact that most nouns show a preference for a countability class. In the present system, we employ the idea of a countability preference particularly in those cases where the context is neutral with regard to countability.

The first step therefore was to determine the countability preferences. We annotated 10,000 German lemmas for their most probable countability class (e.g. *Auto* (car) countable, *Wasser* (water) uncountable). Four trained linguists annotated each noun. Nouns that did not receive a unique annotation were discarded. We furthermore dismissed all nouns that did not show a *class-plausible* ratio of singular and plural occurrences, using the information provided by the RFT. The remaining 4,267 nouns (74% countable, 26% uncountable) were used as prototypical members of their countability class. For these nouns, we counted the co-occurring contexts in the corpora and stored them in the form of a 3-tupel (RFT-POS, TT-POS, lemma) (cf. Table 2).

Context (C)	+count	-count	P(C|+count)
PIAT PRO viel	0	1765	0.0005
KOKOM CONJ wie	327	1200	0.2145
VMFIN VFIN sollen	37	237	0.1376
ART ART einen	246	15	0.9391
PIAT PRO keine	4287	2969	0.5907
…			

Table 2. Example context tuples used by the countability classifier.

We used the *m-estimate* variant of a naïve Bayes classifier (Mitchel, 1997) to determine the probability of a noun being countable given the context (cf. the posterior probabilities given in the last column of Table 2).

For each unseen noun, we calculate a score for being either +COUNT or –COUNT by multiplying the calculated probabilities of occurring contexts, weighted with their frequency. If the normalized score for countability exceeds a defined threshold, the noun is classified as countable.

```
<art source="Neue Zürcher Zeitung" genre="WIRTSCHAFT" date="27.04.1994"
 misc="Nr. 97 31" id="NZZ_1994_04_27_a32"> […]
<para>
<s id="NZZ_1994_04_27_a32_seg5_s13"> […]
<tt_chunk type="PC">
<tok tt_pos="APPR" rft_pos="APPR" rft_lemma="auf" rft_morph="Auf"
 tok_id="NZZ_1994_04_27_a32_seg5_s13_t4">auf</tok>
<tok tt_pos="NN" rft_pos="N" rft_lemma="Anfrage" rft_morph="Reg.Acc.Sg.Fem"
 tok_id="NZZ_1994_04_27_a32_seg5_s13_t5">Anfrage</tok>
</tt_chunk> […]
</s> […]
</para> […]
</art>
```

Figure 1. Abbreviated example of the inline XML format used for the annotation.

If the score is below a second threshold it will be classified as uncountable. A score between those two values results in a classification as *unknown*.

The second classifier bases its classification on the calculated singular/total-ratio of the noun. We trained a decision-tree classifier on all annotated nouns using cross-validation. A singular/total-ratio above 0.997 results in a classification as –COUNT, while a value below 0.98 as +COUNT. Nouns with a value between these two thresholds are classified as *unknown*.

A noun is considered as countable or uncountable if both classifiers reach the same conclusion. Otherwise it is marked as *unknown*.

A first evaluation based on 100 nouns classified as countable and 100 classified as uncountable showed an accuracy of the classifier of 93% in case of countable and 88% in case of uncountable nouns. A more detailed description of the process can be found in Stadtfeld et al. (2009).

3 Indexing and search

3.1 Conversion and indexing

The automatic annotation of our corpora with morphosyntactic features and non-recursive chunks and the training of an accurate countability classifier provide us with all the information necessary to identify and extract PNCs (and also regular PPs). Since the corpora we are currently using already comprise more than 208 million tokens and we are planning to at least double the size of our data base by adding further corpora, we require a search tool that is able to deal with this huge amount of data efficiently.

The (Open) Corpus Workbench (CWB) developed at IMS Stuttgart[1] (Evert, 2005) is well suited to index and query shallow linguistic annotation and has been designed to cope with corpora of more than 100 million words.[2] Moreover, it is also able to index token spans (such as chunks) delimited by XML tags. Therefore, the only minor conversion step necessary to index our inline XML corpus files with CWB consisted in converting the XML annotation of individual tokens into a tab-delimited column format while leaving the XML tags for higher units such as chunks and sentences intact. The information about tokens was encoded by positional attributes, while the information about larger units was encoded using structural attributes. Most importantly, the detailed global IDs defined for all units during the aggregation step were also indexed in CWB in order to enable the unambiguous identification of the extracted constructions for the subsequent manual annotation steps.

We originally planned to create just one big index of all our corpora and to query them all at once in order to make searching less laborious. However, this turned out to be impossible because of RAM limitations. We therefore backed off to indexing whole year volumes of our newspaper corpora separately.

3.2 Searching and extracting PNCs

After indexing the corpora with CWB, we formulated the query shown in (4) to search for PNCs. This query expresses the fact that PNCs form a preposition chunk (PC) which consists of a specific preposition, here exemplified with *an* (on, to), followed by any number of words which are not determiners (i.e., not articles, demonstratives, possessive pronouns, etc.) and finally a regular noun that is both countable and singular.

```
(4) <tt_chunk_type = "PC">
    [(word="an" %cd) &
    (rft_pos="APPR")]

    [(rft_pos!="(ART|…)") &
    (tt_pos!="(ART|…)")]*

    [(tt_pos="NN") &
    (rft_morph!=".*\.Pl\..*")
    & (countability="count")]
    </tt_chunk_type>
```

The list of 23 prepositions that we examine in our study is given in (5). It includes all simple prepositions that typically take an NP complement and also assign case to it.

```
(5) an, auf, bei, binnen,
    dank, durch, für, gegen,
    gemäß, hinter, in, mit,
    mittels, nach, neben,
    ohne, seit, über, um,
    unter, vor, während,
    wegen
```

Examples of prepositions that were excluded are *ab* (from) and *bis* (until), which often occur with a PP or adverbial complement, and the preposi-

1 http://cwb.sourceforge.net/

2 We are also currently looking into the possibility of adopting the search and visualization tool ANNIS2 developed at Humboldt University Berlin and the University of Potsdam (http://www.sfb632.uni-potsdam.de/~d1/annis/). This would be especially useful for the manual inspection of individual examples in later stages of the project. We are currently testing whether ANNIS2 will scale up to very large corpora.

tion *zwischen* (between), which demands a coordinated NP. In general, all prepositions that deviate significantly from the pattern PP = P + NP were excluded.

The query results are exported from CWB as a list of the IDs of all sentences containing at least one PNC. From these lists, reasonably sized working packages can be created, the relevant sentences can be extracted from the inline XML format based on their IDs and can be converted to the format of the annotation tool used for manual annotation (see section 4).

3.3 Evaluation

We performed a small-scale evaluation of our strategy for extracting PNCs in order to determine its effectiveness and quality in terms of precision and recall. For this evaluation, we chose one daily issue of the NZZ randomly: April 3rd, 1997. This issue contains 6,081 sentences and 91,357 tokens. We constructed a gold standard list of PNCs by searching for all occurrences of the prepositions in (5) based only on their word form. All true examples of PNCs were then manually extracted from this large list of 5,304 hits. This yielded a much smaller list of true positives comprising 161 PNCs.[3]

We then used the query expression in (4) to extract all putative PNCs with one of the 23 prepositions from the same NZZ issue based on the automatic morphosyntactic and countability annotation. This resulted in an even shorter list of 56 putative PNCs.

A comparison of the manually and automatically extracted lists of PNCs yielded 27 true positives, 29 false positives, and 134 false negatives. This corresponds to a precision of 48.21% and a recall of 16.77%. The precision of our PNC extraction strategy is satisfactory for our purposes, since irrelevant constructions can still be excluded during the manual annotation phase. The false positives mostly consisted of determinerless nominal complements of prepositions in headlines and coordinations. Since the use of articles follows special rules in these contexts, such examples were excluded in the manual extraction. The low recall is more problematic. It is due to the fact that the countability classifier only classifies nouns for which it has gathered enough contextual information (cf. section 2.3). As discussed in section 1, PNCs are a productive construction and therefore occur with a large number of nouns that the countability classifier has never encountered before. The low recall thus comes from the notorious problem of data sparseness.

This can be shown by extracting a second list of putative PNCs based on the automatic annotation that includes not only nouns classified as countable but also all nouns that were not classified because of a lack of evidence. A comparison of this automatically extracted list with the gold standard results in 143 true positives, 467 false positives, and 18 false negatives, corresponding to a precision of 23.44% and a recall of 88.82%. Recall can thus be increased fivefold, while only halving precision. It might therefore be more sensible to use a classification as uncountable as a knockout criterion rather than to search positively for countable nouns. It is also clear that the coverage of the countability classifier should be improved by training it on larger corpora.

4 Manual annotation

While the automatic annotation steps described in sections 2 and 3 suffice to extract PNCs from corpora, this is only a preliminary task. We are interested in the characteristic properties which distinguish PNCs from PPs, and hence have to annotate further features of PNCs (and corresponding PPs) manually. This step is performed in small batches, since the annotation tool we use cannot deal with large amounts of data and small working packages are also more convenient for the human annotators.

We annotate the relevant constructions with various features such as valency, morphological complexity and etymological status (native vs. borrowed) of the noun and furthermore the semantic interpretation of the respective preposition and noun.

MMAX2 (Müller and Strube, 2006) is employed for manual annotation.[4] It features standoff annotation, which enables us to keep the original corpus and the added annotations separate. Although the annotation tool makes a conversion and preprocessing of the data and the definition of an annotation scheme inevitable, the user has a maximum degree of flexibility in making the tool fit his purposes. Another advantage of MMAX2 is the possibility to create an arbitrary number of independent annotation levels. The annotator is able to add both markables, i.e. spans of tokens, at different levels and pointer relations between the markables.

3 This small number of PNCs shows that huge corpora are indeed required to study such more peripheral constructions.

4 http://mmax2.sourceforge.net/

As a preparatory step, it is necessary to create an MMAX2 project for every batch of sentences, based on the IDs extracted using CWB. Each project consists of several tiers containing the information annotated automatically in the preceding steps, e.g. the information provided by the sentence boundary detection system (level: *sentences*), the TreeTagger (*tt_pos* and *chunks*), the RF-Tagger (*rft*) with the attributes (*rft_pos, rft_lemma, rtf_morph*) and finally the information from the countability classifier (*countability*). New levels for the manual annotation have to be created for the interpretation of the prepositions and nouns (*prep-meaning, noun-meaning*), as well as two levels for the valency of the noun, in order to be able to create pointer relations between the noun and its dependents (*noun-valency, noun-dependents*).

Last but not least, we also define a level at which metadata about the annotation process will be inserted. This will be important to assure completeness of annotation, in particular after reintegrating the manually annotated sentences into the entire corpus. Once the annotation of the PNCs has been completed, we will restart extraction and annotation with ordinary PPs, corresponding to the PNCs we have identified in the first cycle.

5 Conclusion and outlook

The extraction of PNCs is an important yet preliminary step in the determination of the characteristic properties of PNCs.

In this paper, we have shown how automatically annotated data can be used as a basis for extracting the pertinent construction from large corpora. Since we are preparing the data for annotation mining (particularly for clustering and classification), reaching a high recall is as necessary as reaching a high degree of accuracy. Our evaluation has shown some shortcomings of the extraction process in this respect, but a variety of alternative strategies can be considered.

In the current state of affairs, where PNCs have mostly been investigated by looking at individual examples, even an extraction with a relatively low recall facilitates further investigation and will thus be useful to eventually determine the constituting factors of this construction.

References

Keith Allan. 1980. Nouns and countability. *Language* 56(3): 541-567.

Timothy Baldwin, John Beavers, Leonoor van der Beek, Francis Bond, Dan Flickinger, and Ivan A. Sag. 2006. In search of a systematic treatment of determinerless PPs. In Patrick Saint-Dizier, editor, *Syntax and Semantics of Prepositions*. Springer, Dordrecht, pages 163-179.

Christian Chiarcos, Stefanie Dipper, Michael Götze, Ulf Leser, Anke Lüdeling, Julia Ritz, and Manfred Stede. 2008. A flexible framework for integrating annotations from different tools and tagsets. *Traitement Automatique des Langues*. Special Issue Platforms for Natural Language Processing. ATALA, 49 (2).

Florian Dömges, Tibor Kiss, Antje Müller and Claudia Roch. 2007. Measuring the productivity of determinerless PPs. In *Proceedings of the ACL 2007 Workshop on Prepositions*, pages 31-37, Prague, Czech Republic.

Stefan Evert. 2005. *The CQP Query Language Tutorial* (CWB version 2.2.b90). Institut für Maschinelle Sprachverarbeitung, University of Stuttgart.

Tibor Kiss and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4): 485-525.

Tom Mitchell. 1997. *Machine Learning*. McGraw-Hill, New York.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Sabine Braun, Kurt Kohn, Joybrato Mukherjee, editors, *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods. (English Corpus Linguistics Vol. 3)*. Peter Lang, Frankfurt, pages 197-214.

Randolph Quirk, Sidney Greenbaum, Geoffrey Leech, and Jan Svartvik. 1985. *A Comprehensive Grammar of the English Language*. Longman, London.

Helmut Schmid. 1995. Improvements in part-of-speech tagging with an application to German. In *Proceedings of the EACL SIGDAT Workshop*, Dublin, Ireland.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. SMOR: A German computational morphology covering derivation, composition, and inflection. In *Proceedings of LREC 2004*, pages 1263-1266, Lisbon, Portugal.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of COLING 2008*, Manchester, UK.

Tobias Stadtfeld, Tibor Kiss, Antje Müller, and Claudia Roch. 2009. Chaining classifiers to determine noun countability. Submitted to *ACL 2009*, Singapore.

Towards unsupervised learning of constructions from text

Krista Lagus, Oskar Kohonen and Sami Virpioja
Adaptive Informatics Research Centre
Helsinki University of Technology
P.o.Box 5400, 02015 TKK, Finland
{krista.lagus,oskar.kohonen,sami.virpioja}@tkk.fi

Abstract

Statistical learning methods offer a route for identifying linguistic constructions. Phrasal constructions are interesting both from the viewpoint of cognitive modeling and for improving NLP applications such as machine translation. In this article, an initial model structure and search algorithm for attempting to learn constructions from plain text is described. An information-theoretic optimization criteria, namely the Minimum Description Length principle, is utilized. The method is applied to a Finnish corpus consisting of stories told by children.

1 Introduction

How to represent meaning is a question that has for long stimulated research in various disciplines, including philosophy, linguistics, artificial intelligence and brain research. On a practical level, one must find engineering solutions to it in some natural language processing tasks. For example, in machine translation, the translations that the system produces should reflect the intended meaning of the original utterance as accurately as possible.

One traditional view of meaning in linguistics (exemplified e.g. by Chomsky) is that words are seen as basic blocks of meaning, that are orthogonal, i.e., each word is seen as individually conveying totally different properties from all other words (this view has been promoted e.g. by Fodor). The meaning of a sentence, on the other hand, has been viewed as compositional, i.e., consisting of the meanings of the individual words.

Idioms and other expressions that seem to violate against the principle of compositionality (e.g. *"kick the bucket"*) have been viewed as mere exceptions rather than central in language. While such a view might be convenient for formal description of language, and offers a straightforward basis for computer simulations of linguistic meaning, the view has for long been regarded as inaccurate. The problems can also observed in applications such as machine translation. Building a system that translates one word at a time yields output that is incorrect in form, and most often also its meaning cannot be understood.

A reasonable linguistic approach is offered by constructionist approaches to language, where language is viewed as consisting of *constructions*, that is form-meaning pairs.[1] The form component of the construction is not limited to a certain level of language processing as in most other theories, but can as well be a morpheme (*anti-*, *-ing*), a word, an idiom (*"kick the bucket"*), or a basic sentence construction (*SUBJ V OBJ*). The meaning of a sentence is composed from the meanings of the constructions present in the sentence. Construction Grammar is a usage-based theory and does not consider any linguistic form more basic than another. This is well aligned with using data-oriented learning approaches for building wide coverage NLP applications.

We are interested in identifying the *basic information processing principles* that are capable of producing *gradually more abstract representations* that are useful for intelligent behavior irrespective of the domain, be it language or sensory information, and irrespective of the size of the time window being analysed. There is evidence from brain research that the exactly same information-processing and learning principles are in effect in many different areas of the cortex. For example, it was found in (Newton and Sur, 2004) that if during development visual input pathways are re-routed to the region that normally contains auditory cortex, quite typical visual processing and representations ensue, but in this case in the auditory cortical area. The cortical learning al-

[1] For an overview see, e.g., Goldberg (2003).

gorithm and even the model structure can therefore be assumed identical or very similar for both processes. The differences in processing that are seen in the adult brain regions are thus largely due to each region being exposed to data with different kinds of statistical properties during individual growth.

In this article we describe our first attempt at developing a method for the discovery of constructions in an unsupervised manner from unannotated texts. Our focus is on constructions involving a sequence of words and possibly also abstract categories. For model search we apply an information-theoretic learning principle namely Minimum Description Length (MDL).

We have applied the developed method to a corpus of stories told by 1–7 year old Finnish children, in order to look at constructions utilized by children. Stories told by an individual involve entities and events that are familiar to the teller, albeit the combinations and details may sometimes be very imaginative. When spontaneously telling a story, one employs one's imagination, which in turn is likely to utilise one's entrenched representations regarding the world. Of particular interest are the abstract representations that children have—this should tell us about an intermediate stage of the development of the individual.

2 Related work on learning constructions

Constructions as form-meaning pairs would be most naturally learned in a setting where both form and meaning is present, such as when speaking to a robotic agent. Unfortunately, in practice, the meaning needed for language processing is highly abstract and cannot easily be extracted from natural data, such as video. Therefore time consuming hand-coding of meaning is needed and, consequently, the majority of computational work related to learning constructions has been done from text only. A notable exception is Chang and Gurevich (2004) who examine learning children's earliest grammatical constructions, in a rich semantic context.

While learning from text only is unrealistic as a model for child learning, such methods can utilize the large text corpora and discover structure useful in NLP applications. They illustrate that statistical regularities in language form is also involved in learning. Most work has been done within a traditional syntactic framework and thus focuses on learning context-free grammars (CFG) or regular languages. While it is theoretically possible to infer a Probabilistic Context-Free Grammar (PCFG) from text only, in practice this is largely an unsolved problem (Manning and Schütze, 1999, Ch. 11.1). More commonly, applications use a hand crafted grammar and only estimate the probabilities from data. There are some attempts at learning the grammar itself, both traditional constituent grammar and also other alternatives, such as dependency grammars (Zaanen, 2000; Klein and Manning, 2004).

Also related to learning of constructions are the methods that infer some structure from a corpus without learning a complete grammar. As an example, consider various methods that are applied to finding collocations from text. Collocations are pairs or triplets of words whose meanings are not directly predictable from the meanings of the individual words, in other words they exhibit limited compositionality. Collocations can be found automatically from text by studying the statistical dependencies of the word distributions (Manning and Schütze, 1999, Ch. 5).

Perhaps most related to construction learning is the ADIOS system (Solan et al., 2005), which does not learn explicit grammar rules, but rather generalizations in specific contexts. It utilises pseudo-graph data structures and seems to learn complex and realistic contextual patterns in a bottom-up fashion. Model complexity appears to be controlled heuristically. The method described in this paper is similar to ADIOS in the sense that we also use information-theoretic methods and learn a model that extracts highly specific contextual patterns from text. At this point our method is much simpler; in particular, it cannot learn as general patterns. On the other hand, we explicitly optimize model complexity using a theoretically well motivated approach.

3 Learning constructions with MDL

A particular example of an efficient coding principle is the Minimum Description Length (MDL) principle (Rissanen, 1989). The basic idea resembles that of Occam's razor, which states that when one wishes to model phenomenon and one has two equally accurate models (or theories), one should select the model (or theory) that is less complex. In practice, controlling model complexity is essential in order to avoid overlearning, i.e., a sit-

uation where the properties of the input data are learned so precisely that the model does not generalise well to new data.

There are different flavors of MDL. We use the earliest, namely the two-part coding scheme. The cost function to minimize consists of (1) the cost of representing the observed data in terms of the model, and (2) the cost of encoding the model. The first part penalises models that are not accurate descriptions of the data, whereas the second part penalises models that are overly complex. Coding length is calculated as the negative logarithm of probability, thus we are looking for the model $\mathcal{M}^*$:

$$\mathcal{M}^* = \arg\min_{\mathcal{M}} L(\text{corpus}|\mathcal{M}) + L(\mathcal{M}). \quad (1)$$

The two-part code expresses an optimal balance between the specificity and the generalization ability of the model. The change of cost can be calculated for each suggested modification to the model.

Earlier this kind of MDL-based approach has been applied successfully in unsupervised morphology induction. For example, the language-independent method called Morfessor (Creutz and Lagus, 2002; Creutz and Lagus, 2007) finds from untagged text corpora a segmentation for words into morphemes. The discovered morphemes have been found to perform as good as or better than linguistic morphemes or words as tokes for language models utilized in speech recognition (Creutz et al., 2007). It is therefore our hypothesis that a similar MDL-based approach might be fruitfully applied on the sentence level as well, to learn a "construction inventory" from plain text.

3.1 Model and cost function

The constructions that we learn can be of the following types:

- word sequences of different lengths, e.g., `went to`, `red car`, `and`
- sequences that contain one *category*, where a category refers simply to a group of words that is expected to be used within this sequence, i.e. `went to buy [X]`, `[X] was`.

If only the former kind of structure is allowed, the model is equivalent to the Morfessor Baseline model(Creutz and Lagus, 2002), but for sentences consisting of words instead of words consisting of letters. Initial experiments with such a model showed that while the algorithm finds sensible structure, the constructions found are very redundant and therefore impractical and difficult to interpret. For these reasons we added the latter construction type. However, allowing only one category is merely a first approximation, and later we expect to consider also learning constructions with more than one abstract category.

The coding length can be calculated as the negative logarithm of the probability. Thus, we can work with probability distributions instead. In the likelihood we assume that each sentence in the corpus is independent and that each sentence consists of a bag-of-constructions:

$$P(\text{corpus}|\mathcal{M}) = \prod_{i}^{N} P(s_i|\mathcal{M})$$

$$P(s_i|\mathcal{M}) = \prod_{j}^{M_i} P(\omega_{ij}|\mu_{ij}, \mathcal{M}) P(\mu_{ij}|\mathcal{M})$$

where s_i denotes the i:th sentence in the corpus of N sentences, M_i is the amount of constructions in s_i, μ_{ij} denotes a construction in s_i and ω_{ij} is the word that fills the category of the construction (if the construction has a category, otherwise that probability = 1). The probabilities $P(\mu_{ij}|\mathcal{M})$ and $P(\omega_{ij}|\mu_{ij}, \mathcal{M})$ are multinomial distributions, whose parameters need to be estimated.

When using two part codes the coding of the model may in principle utilize any code that can be used to decode the model, but ideally the code should be as short as possible. The coding we use is shown in Figure 1. We apply the following principles: For bounded integer or boolean values (fields 1, 2.1, 2.3, 2.4 and 4.1 in Figure 1) we assume a uniform distribution over the possible values that the parameter can take. This yields a coding length of $\log(L)$, where L is the amount of different possible values. For the construction lexicon size (field 1), L is the number of n-grams in the corpus and its coding length is therefore constant.

When coding words (fields 2.2 and 4.2) we assume a multinomial distribution over all the words in the corpus, and the parameters are estimated from corpus frequencies. Thus the probability of construction lexicon units (field 2.2) is given by:

$$P(\text{words}(\mu_k)) = \prod_{j}^{W_k} P(w_{kj}), \quad (2)$$

1. Number of constructions	2. Constructions μ_i:	3. Construction counts	4. Categories:
	2.1. length(μ_i)		4.1. number of words
	2.2. words in μ_i		4.2. words
	2.3. has category?		4.3. word counts
	2.4. category position		

Figure 1: Coding scheme for the model.

where W_k is the number of words in construction μ_k and $P(w_{kj})$ the probability of a word. The category words (field 4.2) are coded in a similar manner.

We also need to encode the parameters for the multinomials $P(\mu_{ij}|\mathcal{M})$ and $P(\omega_{ij}|\mu_{ij}, \mathcal{M})$. We do this by encoding the corresponding counts (fields 3 and 4.3), from which the probabilities can be calculated. We use the following reasoning: If there are M different construction or word types and the sum of their counts is K, then there are $\begin{pmatrix} K-1 \\ M-1 \end{pmatrix}$ ways of choosing M positive integers so that they sum up to K. Thus the coding length is the negative logarithm of

$$P(\text{count}(\mu_1), .., \text{count}(\mu_M)) = 1/\begin{pmatrix} K-1 \\ M-1 \end{pmatrix}. \quad (3)$$

3.2 Search algorithm

Because we are optimizing both model parameters and model size at the same time, standard probabilistic parameter estimation methods, such as Expectation-Maximization, cannot be used. Instead we use an incremental algorithm for optimizing the cost function as follows: At all times we maintain a certain analysis of the corpus and try to improve it. For a given analysis it is possible to estimate the maximum likelihood parameters for $P(\omega_{ij}|\mu_{ij}, \mathcal{M})$ and $P(\mu_{ij}|\mathcal{M})$ and then calculate the cost function for that model.

The optimization proceeds with the following steps: (1) Initialize the analysis so that each word is a construction by itself and there exist no other constructions. (2) Generate all possible constructions of length ≤ 6 from the corpus. For those constructions that exist more than 10 times in the corpus, calculate the likelihood ratio. Since the likelihood side of the optimization is completely local one can calculate the change in likelihood that one would get from modeling a set of sentences using a certain construction, compared to the initial analysis. (3) In the descending order of likelihood ratios, apply the construction to all sentences where applicable. Then calculate the value of the cost function. If the change improved the cost, accept it, otherwise discard the change. Finally, proceed with the next construction.

4 Experiments

We applied our MDL-based model to a corpus consisting of stories told by Finnish children. The are several reasons for this choice of data. If one is interested in underlying cognitive processes and their development, it may be more fruitful to look at the outputs of a cognitive system in the middle of its development rather than modeling the outputs of the fully developed system. Because the data that children hear is produced by adult systems, some of it is likely to be discarded by children by means of attentional selection, and one cannot easily know which part. This problem is avoided by only looking at data that is known to be represented by the children, that is, produced by them. From the practical point of view, as we have no means of quantitative evaluation, we want to apply the method to such a data that should have many frequent and simple constructions to observe.

4.1 Corpus and preprocessing

The corpus contains 2642 stories told by children to an adult—typically a day care personnel or a parent—who has written the story down exactly as it was told, without changing or correcting anything. A minority of the stories were told together by a pair or by a group of children. The children ranged from 1 to 7 years. The story markup contains the age and the first name(s) of the storyteller(s). The stories contain a lot of spoken-language word forms. For a more extensive description of the corpus, see (Klami, 2005).

A story told by Oona, 3 years: *Mun äitin nimi on äiti. Mun iskän nimi on iskä. Iskä tuli mun kanssa tänne. Mun nimi on Oona. Jannen nimi on Janne.* A story told by Joona, 5 years and

11 months: *Dinosaurus meni kauppaan osti sieltä karkkia sitten se meni kotiin ja söi juuston. Sitten se meni lenkille ja se tappoi pupujussin iltapalaksi ja sitten se meni uudestaan kauppaan ja se ei näkenyt mitään siellä kun kauppa oli kiinni.*

The stories are preprocessed as follows: Story mark-up containing headers etc. is removed, any punctuation is replaced with a symbol # and the story is divided into sentences. After removal of story mark-up the total number of sentences in the corpus is 36,542. The number of word tokens is 244,274 and word types 24,242. Each sentence is then given as input for the construction learner.

4.2 Results

Figure 2 shows the most frequent constructions that the algorithm has discovered. One can see that the frequent constructions found by the algorithm are good, in the sense that they are not random frequent strings, but often meaningful constructions. An especially nice example of a construction found is `olipa kerran [X]`, which is the archetypical way of beginning a fairy tale in Finnish (*once upon a time there was a ...*). The prominence of `ja sitten` is caused by many stories following a pattern where the child explains some event, then uses `ja sitten` to move on to the next event and so on. The algorithm has discovered one piece of this pattern. We also see that the algorithm has discovered that the spoken language forms of `sitten` (*then*)—`sit`, `sitte` and `sitt`—are similar.

When looking at the categories, it can be seen that they are sometimes overly general. E.g., `meni metsään` and `meni #` are analysed as `meni [X]`, where in the former case `[X]` is the argument of the verb, and in the latter the verb takes no arguments, but happens to be at the end of a sentence. However, in many cases the discovered categories appear to consist of one or a few semantic or part-of-speech categories. E.g., `söi [X] #` (*ate [X] #*) contains mostly edible arguments `banaania` (*banana*), `mansikkaa` (*strawberry*), `jäniksen` (*a rabbit*) or a pronoun `hänet` (*him/her*), `ne` (*them*).

Whereas these frequent constructions are fairly good, the analyses of individual sentences generally leave much available structure unanalysed. Consider the analysed sentence: `että hirveä hai tuli niitten [perään {X → ja}] [söi {X → ne}] #` (*that terrible shark came them [after {X → and}] [ate {X → them}] #*). We can see that most of the sentence is not analysed as any abstract construction. Looking at the corpus, we can see possible constructions that the algorithm does not discover. E.g., constructions such as `[X] hai`, where the category contains adjectives or `hai [X]` where the category contains an action verb. Note also that both constructions could not currently be used at the same time, but one would have to choose either.

5 Discussion

As this is our first attempt at learning a construction inventory, there are still many things to consider. Regarding learning of the model, one a more local updating step, in addition to the current global update, would be needed. Also, the algorithm should consider merging categories that have partially overlapping words.

Currently the model structure allows only a very restricted set of possible constructions, namely exact phrases and partially filled constructions that have exactly one abstract category that can be filled by one word. It is later possible to relax both constraints, and allow a category to be filled by several consecutive words, as well as allowing many abstract categories per construction. However, adding such abstraction capability will increase the search space of possible models quite radically, bringing the complexity close to learning a PCFG from unannotated text.

Starting simple is thus prudent: we wish to ensure learnability of the model. Moreover, we wish to identify the simplest possible approach and model structure that can account for interesting and complex phenomena, when applied throughout a corpus. A possible alternative to PCFGs would be to keep the constructions simple, but allow them to overlap each other.

Our goals include also applying the found constructions to NLP applications such as machine translation. The current statistical machine translation systems solve the problems of non-compositionality by translating a longer sequence of words (phrase) at a time. However, finding the phrase pairs is usually quite heuristic, and the phrases do not include any abstract categories. Even a reasonably simple algorithm for finding more abstract constructions should help alleviate the data sparsity problems. Applying construction learning into applications is also useful as a way of evaluating the results, as there is no "gold stan-

Most frequent constructions of two words		
Freq.	**Form**	**Category words (freq.)**
891	hän [X]	meni (68), oli (50), lähti (32), löysi (29), otti (19)
	he [X]	*went, was, left, found, took*
885	ja sitten	
	and then	
798	[X] on	se (82), hän (24), täällä (20), tässä (20), nyt (17)
	[X] is	*it, he/she, here, here, now*
768	meni [X]	metsään (33), ulos (33), sinne (30), # (25), nukkumaan (18)
	went [X]	*(into the) forest, outside, there, #, (to) sleep*
694	sit [X]	se (302), ne (81), kun (20), hän (17), # (12)
	then [X]	*it, they, when, he/she, #*
Most frequent constructions of three words		
632	ja [X] se	sitten (303), sit (155), sitte (109), sitt (18), kun (5)
	and [X] it	*then, then, then, then, when*
337	[X] se meni	sitten (125), sit (66), sitte (58), ja (35), kun (14)
	[X] it went	*then, then, then, and, when*
245	olipa kerran [X]	pieni (8), tyttö (7), yksi (6), koira (6), hiiri (5)
	once (upon a time) there was (a) [X]	*little, girl, one, dog, mouse*
235	ja [X] ne	sitten (129), sit (37), sitte (28), kun (6), niin (5)
	and [X] they	*then, then, then, when, so*
197	ja [X] tuli	sitten (91), se (9), sinne (6), ne (4), niistä (3)
	and [X] came	*then, it, there, they, (of) them (be-)*

Figure 2: The most frequent two- and three word constructions with their five most frequent category words.

dards" for direct automatic evaluation.

6 Conclusions

We share the intuition found in cognitive linguistics in general, that constructions are able to capture something essential about the cognitive representations that are also the basis of our actions and situatedness in the world.

It is our hope that the study of constructions, and the endeavour of learning them from corpora and perhaps later from richer behavioral and perceptual contexts might eventually provide a new opening in the field of modeling both language and cognition.

References

N. Chang and O. Gurevich. 2004. Context-driven construction learning. In *Proc. CogSci 2004*, Chicago.

M. Creutz and K. Lagus. 2002. Unsupervised discovery of morphemes. In *Proc. Workshop on Morphological and Phonological Learning of ACL'02*, pages 21–30, Philadelphia, Pennsylvania, USA.

M. Creutz and K. Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1).

M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke. 2007. Morph-based speech recognition and modeling of out-of-vocabulary words across languages. *ACM Transactions on Speech and Language Processing*, 5(1).

A. E. Goldberg. 2003. Constructions: a new theoretical approach to language. *Trends in Cognitive Sciences*, 7(5):219–224.

M. Klami. 2005. Unsupervised discovery of morphs in children's stories and their use in self-organizing map -based analysis. Master's thesis, University of Helsinki, Department of General Linguistics, Helsinki, Finland.

D. Klein and C. D. Manning. 2004. Corpus-based induction of syntactic structure: Models of dependency and constituency. In *Proc. ACL 2004*, pages 478–485, Barcelona, Spain.

C. D. Manning and H. Schütze. 1999. *Foundations of Statistical Language Processing*. The MIT Press.

J.R. Newton and M. Sur. 2004. Plasticity of cerebral cortex in development. In G. Adelman and B.H. Smith, editors, *Encyclopedia of Neuroscience*. Elsevier, New York, 3rd edition.

J. Rissanen. 1989. *Stochastic Complexity in Statistical Inquiry*. World Scientific Publ. Co., New Jersey.

Z. Solan, D. Horn, E. Ruppin, and S. Edelman. 2005. Unsupervised learning of natural languages. *Proceedings of the National Academy of Sciences*, 102(33):11639–11634.

M. van Zaanen. 2000. Bootstrapping syntax and recursion using alignment-based learning. In P. Langley, editor, *Proc. ICML 2000*, pages 1063–1070. Stanford University, Morgan Kaufmann Publishers.

Using collocation-finding methods to extract constructions and to estimate their productivity

Kadri Muischnek
University of Tartu
Tartu, Estonia
Kadri.Muischnek@ut.ee

Heete Sahkai
Institute of the Estonian Language
Tallinn, Estonia
Heete.Sahkai@eki.ee

Abstract

This paper describes first an attempt to identify constructions as colligational patterns and to extract instances of certain constructions from a large text corpus of Estonian using collocation extraction methods. Focusing then on the productivity of constructions, authors use three different methods for its estimation: rankings by a collocation association measure, measures of (morphological) productivity and semantic analysis. The methods proved to be consistent, suggesting that collocational analysis can be used as an indicator of productivity.

1 Introduction

This paper reports an attempt to use collocation-finding methods as a way of detecting and extracting partially schematic constructions and of estimating their productivity. It was intended as an exploratory study aimed at determining whether such methods could provide useful tools for these aspects of constructional research. If the results turn out to be promising, more sophisticated computational methods and theoretical concepts will have to be applied to different types of constructions in order to further test and elaborate these tools[1].

We tested the methods on complex predicate (CP) patterns containing the verb *minema* 'go'. *Minema* (as GO-verbs in general) is a polysemous verb which occurs in a variety of constructions, including a considerable number of multi-word expressions (Muischnek 2006:53) and various more or less productive CP patterns. The material should thus be suitable both for the construction detection and extraction tasks and for the productivity estimation test.

We started by applying multi-word expression (MWE) extraction methods in order to find the colligations (i.e. co-occuring grammatical categories) of *minema*, which we then ranked on the basis of a lexical association measure. Our aim was to test whether these methods could be used to identify constructions.

Next, we examined more closely two CP patterns containing *minema*, with the aim of testing collocation-finding methods as a way of extracting particular constructions and of determining their productivity. We again applied collocation-finding methods to extract the instantiations of these patterns from the corpus and an association measure to rank the list of the extracted pairs in order to test the hypothesis that the rankings can be used to estimate the productivity of the examined patterns.

The paper is structured as follows. Section 2 describes the methods and the results of the colligation and construction extraction tasks and of the collocational analysis. Section 3 discusses the possibility of using collocational analysis as a tool for estimating the productivity of constructions. Section 4 presents the summary and some further perspectives of research.

2 Colligation and construction extraction and collocational analysis

For our experiments we used a morphologically tagged corpus of Estonian, consisting in equal parts of newspaper, fiction and scientific texts. In our corpus every word-form in text has been annotated for its lemma and part of speech; nouns have been annotated for their number and case, verbs for their relevant grammatical categories. In the colligation and construction extraction tasks, candidate word pairs were obtained from co-occurences of words separated by up to three intervening tokens excluding punctuation and not crossing sentence boundaries.

In order to divide or rank the candidate word pairs as MWEs and productively formed word combinations, various lexical association measures are used. Pavel Pecina (2005, 2008) and Pe-

[1] We thank an anonymous reviewer for valuable suggestions to this effect.

cina and Schlesinger (2006) list more than 80 such measures. Still, one can't name the best-performing measure as these tasks depend heavily on data, language and the type of MWE (Pecina and Schlesinger 2006; Evert 2008). For a similar task of ranking instantiations of particular constructions Anatol Stefanovitsch and Stefan Th. Gries (2003) use Fischer's exact test, which makes no distributional assumptions nor requires any particular sample size, but is computationally very expensive. In our experiments we applied t-score as a proven association measure (Evert 2008: 22) which performed well in our initial test studies and should be sufficient for the aims of an exploratory study. We used it for two tasks: 1) for ranking colligational patterns, i.e. co-occurences of a lexical item (verb *minema* 'go') and a grammatical category (2.1), and 2) for ranking word pairs (instances of these colligations) (2.2).

2.1 Colligation extraction

To begin with, we applied MWE extraction methods in order to find colligations of the verb *minema* 'go'. The top 7 colligations of the verb *minema* ordered using the t-score lexical association measure are presented in Table 1.

Word-class and grammatical category	Total in corpus	In colligation with *minema*	t-score
verb supine	97893	4240	40,999
noun sg aditive	83992	3809	39,886
noun sg illative	37392	1780	27,973
noun sg allative	188813	4265	18,929
noun sg translative	188770	3924	14,302
verb imperf 1 sg	59862	1148	5,541256
verb present 1 sg	51441	960	4,351668

Table 1. The most significant colligations (grammatical collocations) of the verb *minema* according to the t-score association measure

The most significant colligation of the verb *minema* 'go' is a verb in supine form followed by four nominal case forms, namely aditive, illative, allative and translative. On the 6. and 7. positions there are two verbal categories, namely imperfect first person singular and present first person singular. It is noteworthy that there is a considerable decrease in t-score value after the noun in translative case.

We also calculated three other association scores: local-MI (Mutual Information), log-likelihood and chi-squared. All these association measures ranked supine as the most significant colligation of *minema*; chi-squared and local-MI also placed the translative case at the 5. position, whereas log-likelihood ranked it as 4. So there were no big differences at the very top of the lists of colligations ranked by different association measures.

From the perspective of the detection of constructions, these results seem interesting: each colligation in the top of the list represents several constructions, among which are the principal complementation structures of *minema* as well as various CP constructions. Furthermore, as we analyzed manually the results of the extraction of the word pairs instantiating single colligations (cf. 2.2), we were able to identify some new constructions that are not recorded in the dictionaries. It seems thus worthwhile to pursue the study of using lexical association measures to estimate colligational strength and to identify partially schematic constructions.

Another way to use collocation extraction methods for the identification of constructions would be to detect families of multi-word expressions and look for less-frequent examples of the same pattern, as frequently used instances of constructions tend to freeze and become multi-word expressions, and conversely, constructions tend to emerge as extensions of multi-word expressions.

Three of the nominal cases in Table 1 – aditive, illative and allative - are the cases of spatial destination, and could be expected to co-occur with a verb meaning 'go'. So the results presented in Table 1 guided our attention to two grammatical categories co-occuring with the verb *minema* – the supine form of a verb and the translative case form of a noun.

2.2 Construction extraction

In the present study, two constructions (and fixed expressions related to them) formed with *minema* are analyzed, with the aim of testing 1) collocation finding methods as a tool for extracting constructions, and 2) collocational analysis as an indicator of productivity (cf. section 3). Both constructions are CP constructions with an inceptive meaning. In the first one (1) *minema* combines with a supine form of the verb[2] and in the second one (2) with a translative[3] noun, which is usually derived from a verb.

[2] The supine, alternatively termed "ma-infinitive", functions as an infinitive and occurs i.a. as the complement of aspectual and modal verbs, e.g. 'start', 'must'.

[3] Estonian has a system of 14 nominal cases; the translative is primarily the case of the end-state complements of change-of-state verbs, e.g. *värvib kollaseks* 'paints yellow'.

(1) Maja läheb põlema
house-NOM[4] goes burn-SUP
'The house takes fire'
(2) Nüüd läheb tantsuks
now goes dance-TRANSL
'Now the dancing starts'

The two patterns are illustrated in "The defining dictionary of standard Estonian" (1988-2008) with a more or less equal number of examples: the supine pattern with 13 examples and the translative pattern with 10 examples. However, neither the dictionary entry nor our intuition as native speakers permits us to determine whether these examples constitute a series of similar fixed expressions or exemplify productive patterns, nor what could be the degree of productivity of these eventual patterns. They provided thus interesting material for testing productivity measures, since productivity is most difficult to determine in the intermediate cases between idiosyncracy and generality (Boas 2008).

To examine these constructions in detail, we extracted word pairs consisting of *minema* and a supine verb form or a translative noun form from our corpus. The extracted lists of word pairs were examined by hand and tagged as fixed expressions, instances of constructions relevant for the present study, and (occasional) combinations not relevant for the present study.

Using the described methods we extracted 512 word pairs consisting of the verb *minema* and a supine form of a verb and 1151 word pairs consisting of *minema* and a translative noun. Most of the *minema* + supine combinations represented either the construction exemplified in (1) or the purposive construction, e.g. *läks sööma* 'went eat-SUP; went to eat'.

The *minema* + translative pattern comprises a more diverse set of constructions. In addition to the construction exemplified in (2), the translative noun in combination with *minema* can be used for expressing end-state in a change-of-state construction, e.g. *läks kooli õpetajaks* 'went school-ADIT teacher-TRANSL; became a teacher at school', timespan, e.g. *läks nädalaks reisile* 'went week-TRANSL journey-ALL; went to a journey for a week', and purpose, e.g. *läks emale abiks* 'went mother-ALL help-TRANSL; went to help his mother'.

As the next step, we applied the t-score association measure to rank the list of extracted pairs. We hypothesized that the association measure should give us ranked lists of word pairs with the fixed expressions (both idiomatic and conventionalized) on the top and the productively combined instances of constructions scattered along the whole list, but below the fixed expressions. The tables 2-5 present the distribution of the instances of the constructions in the lists. For comparison, we also present the rankings of the fixed expressions of the same form, which do not all represent the examined constructions. As the number of candidate expressions extracted from the corpus as well as the number of instances of the respective constructions was different for the two constructions, the rank sizes are different.

rank	t-score	cumulative nr of expressions	cumulative %
1->25	16,423-4,044	2	7,4
1->50	2,604	6	22,2
1->100	1,658	12	44,4
1->150	1,244	18	66,7
1->200	0,984	18	66,7
1->250	0,920	19	70,4
1->300	0,840	22	81,5
1->350	0,695	24	88,9
1->400	0,423	24	88,9
1->450	-0,187	27	100
1->500	-3,268	27	100

Table 2. Distribution of the *minema* + supine inceptive construction (512 candidate pairs altogether)

rank	t-score	cumulative nr of expressions	cumulative %
1->25	16,423-4,044	9	60
1->50	2,604	12	80
1->100	1,658	13	86,7
1->150	1,244	13	86,7
1->200	0,984	14	93,3
1->250	0,920	14	93,3
1->300	0,840	14	93,3
1->350	0,695	14	93,3
1->400	0,423	14	93,3
1->450	-0,187	14	93,3
1->500	-3,268	15	100

Table 3. Distribution of *minema* + supine fixed expressions (512 candidate pairs altogether)

[4] abbreviations: ADIT-aditive case; ALL-allative case; NOM-nominative case; SUP-supine; TRANSL-translative case

rank	t-score	cumulative nr of expressions	cumulative %
1->50	11,931-2,930	0	0
1->100	1,960	7	5,8
1->200	1,357	18	15
1->300	0,984	33	27,5
1->400	0,984	46	38,3
1->500	0,968	66	55
1->600	0,952	79	65,8
1->700	0,904	91	75,8
1->800	0,823	98	81,7
1->900	0,632	104	86,7
1->1000	0,099	120	100

Table 4. Distribution of *minema* + translative inceptive construction (1151 candidate pairs altogether)

rank	t-score	cumulative nr of expressions	cumulative %
1->50	11,931-2,930	4	44,4
1->100	1,960	6	66,7
1->200	1,357	7	77,8
1->300	0,984	7	77,8
1->400	0,984	8	88,9
1->500	0,968	8	88,9
1->600	0,952	9	100
1->700	0,904	9	100
1->800	0,823	9	100
1->900	0,632	9	100
1->1000	0,099	9	100

Table 5. Distribution of *minema* + translative fixed expressions (1151 candidate pairs altogether).

Comparing the tables 3 (*minema* + supine fixed expressions) and 2 (*minema* + supine inceptive construction), we can see that our hypothesis – by ordering the list of candidate expressions using association measures one can get a ranked list with the fixed expressions at the top and instances of constructions scattered all over the list – generally holds. In a list ordered by t-score the first 200 candidate pairs include 93% of the fixed expressions consisting of the verb *minema* and a verb in supine form. Still, the instances of the inceptive *minema* + supine construction show a tendency to cumulate in the first half of the ranked list (the first third of the ranked list contains 67% of the instances of this construction). The fixed expressions consisting of *minema* and a translative noun (Table 5) behave quite like the fixed expressions consisting of *minema* and a verb in supine form: they are concentrated in the first half of the ranked list. But the instances of the *minema* + translative inceptive construction (Table 4) do not occur in equal proportions in the very top of the ranked list.

We hypothesized that the distribution of the instantiations of a pattern in a list ranked by a lexical association measure could indicate the productivity of the pattern: if the expressions instantiating the pattern cluster in the top of the list, the pattern is more likely to consist of conventionalized expressions; and if the instantiations are scattered along the whole list, or concentrate in the end of the list, the pattern is more likely to be productive. We will attempt to test this hypothesis in the next section.

Although ranking the word pairs using a lexical association measure helps us to distinguish between MWEs and productively formed instances of constructions, we are not able to distinguish automatically the instances of different constructions formed by the same grammatical pattern. Using semantic information might help, e.g. classifying the verbs participating in the *minema* + supine pattern as agentive/non-agentive could help us to distinguish between the inceptive and purposive constructions (cf. section 3 for the semantic properties of the inceptive construction). Another method worth experimenting would be to use context (either lexical or grammatical) entropy (cf. Pecina and Schlesinger 2006). Still, even at the present stage, the method is of great help for the linguistic analysis of these constructions.

3 Collocational analysis as an indicator of productivity

In order to test the possibility of using collocational analysis as an indicator of productivity, we compared the results in tables 2-5 with two productivity measures proposed by Harald Baayen, and conducted a semantic analysis, since productivity has also been found to correlate with semantic coherence (Bardðal 2008, Bybee and Eddington 2006). If the results turn out to be interesting, more sophisticated productivity measures (e.g. the LNRE models described by Baayen 2001) will have to be compared with results obtained with different association measures (cf. section 2) in order to further develop the method.

Productivity measures. To measure the productivity of the examined constructions we used two methods originally suggested by Harald

Baayen (ms:6-18) for measuring morphological productivity: realized productivity and potential productivity. Realized productivity measures the size of the category and is estimated by the type count of the construction: how many different instances of a construction are there in the corpus. Potential productivity estimates the growth rate of the category and is calculated dividing the *hapax legomena* of the construction by the total number of its tokens in the corpus.

The results of applying the measures to the two inceptive constructions and, for comparison, to the MWEs of the form *minema* + supine, are presented in table 6.

	transl. cn	supine cn	MWEs
realized productivity (types)	120	27	15
hapaxes	83 (~68%)	7 (~25%)	0
tokens	198	229	1153
potential productivity	0,41	0,03	0

Table 6. Productivity measures of the *minema* + translative and *minema* + supine inceptive constructions, and of the MWEs of the form *minema* + supine

The results obtained by applying the productivity measures seem to correlate with the data of the t-score-ranked lists presented in tables 2-5: the *minema* + supine inceptive construction (Table 2) with lower productivity measures shows a tendency to cumulate in the first half of the ranked list, while the instances of the *minema* + translative construction (Table 4), which has higher productivity measures, do not appear in equal proportions in the very beginning of the ranked list, i.e. there are no instances of this construction with high t-score values.

A semantic analysis. We also analyzed the data from the point of view of semantic coherence, which has been found to increase productivity (Bybee and Eddington 2006), especially in constructions of lower type frequency (Bardðal 2008:44). Conversely, constructions with higher type frequency are more productive if they are less coherent, i.e. more schematic (Bardðal 2008:45).

The verbs occurring in the supine pattern form three similarity groups comparable to the ones described by Bybee and Eddington (2006). 16 verbs express a physical state or process (burn, grow, boil, rot…), 6 verbs express undirected movement of non-agentive subjects (move, roll, rotate…), and 3 verbs express physical or verbal conflict (fight, argue…). In fact, the expressions in the second group may be an extension of the first group because they cannot take goal or source adjuncts, suggesting that they construe movement as a state. The expressions in the third group specifically express involuntary involvement in a conflict. Thus all three groups express non-agentive events, possibly as a result of the fact that the inceptive pattern seems to derive from non-causative change-of-state expressions. The first two groups are represented each by a frequent MWE as well as by 1 and 3 hapaxes, respectively. The third group contains neither a central member nor any hapaxes. Two combinations that are not directly related to these groups are a MWE *juhtima minema* 'take the lead' (lit. 'go to lead'), and a hapax *venima minema* 'start to drag on' (lit. 'go to drag on').

The nouns occurring in the translative pattern constitute a radial family-resemblance category with a central group of 34 nouns expressing physical or verbal conflict. This group contains the three most frequent nouns in the pattern (fight 9, war 8, struggle 6). The other nouns occurring in the pattern are variously linked to this group: 12 nouns denote communication or verbal or vocal activity (discussion, screaming…); 24 nouns express various other types of collective or interactive events or activities (turmoil, party, competition, strike, business, sex…). The remaining 27 nouns denote various activities and accomplishments that are not inherently collective but mostly had multiple participants in the context (work, building, stealing), although in a few cases a single participant was involved. Thus the translative expressions can also be subsumed under a common denominator, "social interaction", but this is again not so much a function of the meanings of the nouns that occur in them as an effect of constructional semantics. An unrelated expression in the data is *sajuks minema* 'go rain-TRANSL', which can be seen to connect to the pattern that is probably at the origin of the inceptive pattern, namely a change-of-state pattern involving the verb *minema* and a translative adjective. Unlike the supine pattern, the translative pattern is represented neither by single high-frequency expressions nor by MWEs.

In conclusion, both patterns are to some extent coherent on the constructional level, but the nouns occurring in the translative pattern are more diverse than the verbs occurring in the supine pattern. The higher coherence of the supine pattern together with its lower type count and the higher t-score values of its instances suggests that the productivity of the pattern consists in analogical extensions (cf. Bardðal 2008:2-3)

based on conventionalized expressions. The lower coherence and the higher type count of the translative pattern in turn is consistent with the lower t-score values of its instances, indicating that it is more productive.

The correlation between the results of the collocational analysis and the independent indicators of productivity suggests that it might indeed be possible to estimate the productivity of a construction by inspecting the distribution of its instantiations in a list ranked by some lexical association measure.

4 Summary of the results and further perspectives of research

Our study suggests that collocation-finding methods could be developed into useful tools for constructional research.

In the first task, MWE extraction methods and the t-score association measure identified those colligations of the verb *minema* that were the most relevant for the detection of constructions: each of them represented several constructions that were among the principal complementation and CP patterns of *minema*, but also some patterns that were not recorded in dictionaries.

In the next task, we used collocation-finding methods to extract word pairs instantiating two of these colligations, with the aim of finding the instantiations of two specific constructions. The results required additional manual analysis but provided the data we needed.

Finally, we applied the t-score association measure to the extracted word pairs and found that the distribution of the instances of the examined constructions in the respective t-score-ranked lists correlated with some independent indicators of productivity and contributed to the estimation of the nature and degree of the constructions' productivity.

It remains for future research to develop these methods further by testing them on different types of constructions, by comparing the results of different association measures, and by applying more sophisticated productivity measures.

Acknowledgements

This study was funded by the targeted financing projects SF0180078s08 and SF0050023s09 of the Estonian Ministry of Education and Research.

References

Baayen, Harald R. (ms.) Corpus linguistics in morphology: morphological productivity. http://www.ualberta.ca/~baayen/publications/BaayenCorpusLinguistics2006.pdf

Baayen, Harald 2001. Word Frequency Distributions. Dordrecht/Boston/London: Kluwer Academic Publishers.

Bardðal, Jóhanna 2008. Productivity. Evidence from Case and Argument Structure in Icelandic. Amsterdam, Philadelphia: John Benjamins Publishing Company.

Boas, Hans C. 2008. Determining the structure of lexical entries and grammatical constructions in Construction Grammar. *Annual Review of Cognitive Linguistics 6*, 113-144.

Bybee, Joan and David Eddington 2006. A usage-based approach to Spanish verbs of 'becoming'. *Language* 82:2, 323-355.

The Defining dictionary of standard Estonian = Eesti kirjakeele seletussõnaraamat, Tallinn, 1988–2008

Evert, Stefan 2008. Corpora and collocations. - A. Lüdeling and M. Kytö (editors), *Corpus Linguistics. An International Handbook*, article 58. Mouton de Gruyter, Berlin. [extended manuscript: http://purl.org/stefan.evert/PUB/Evert2007HSK_extended_manuscript.pdf]

Muischnek, Kadri 2006. Eesti keele verbikesksete püsiühendite nominaalsetest komponentidest.In: *Keel ja arvuti. Tartu Ülikooli üldkeeleteaduse õppetooli toimetised 6*, pp 51-71.

Pecina, Pavel 2005. An Extensive Empirical Study of Collocation Extraction Methods. *Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL 2005), Student Research Workshop pp 13-18, Ann Arbor, Michigan, June 2005.*

Pecina, Pavel 2008. A Machine Learning Approach to Multiword Expression Extraction. *Proceedings of the LREC 2008 Workshop Towards a Shared Task for Multiword Expressions (MWE 2008), pp 54-57, Marrakech, Morocco, May 2008*

Pecina, Pavel and Pavel Schlesinger 2006. Combining Association Measures for Collocation Extraction. *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp 651-658, Sydney, July 2006.*

Stefanowitsch, Anatol and Stefan Th Gries 2003. Collostructions: Investigating the interaction of words and constructions. *International Journal of Corpus Linguistics* 8:2, 209-243.

A corpus-based tool for helping writers with Swedish collocations

Robert Östling
School of Computer Science
and Communication
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
robost@gmail.com

Ola Knutsson
School of Computer Science
and Communication
Royal Institute of Technology
SE-100 44 Stockholm, Sweden
knutsson@csc.kth.se

Abstract

A prototype of a corpus-based tool to help writers use collocations in written Swedish is presented. Unlike similar tools for English, fully parsed and lemmatized text is used, to accommodate the Swedish language with its inflections and varying word order.

1 Introduction

The task of extracting collocations from a corpus has been investigated at length by several authors. Evert (2005) and Pearce (2002) discuss and evaluate much of the previous research. However, machine-assisted methods for *changing* collocation usage have received less attention. There have been a few recent attempts to construct practical tools intended to improve collocation usage in English (Shei and Pain, 2000; Park et al., 2008; Chang et al., 2008; Futagi et al., 2008), but none for Swedish.

This paper presents the results of a project attempting to construct such a tool for Swedish.

2 Design

The overall goal of the tool is to help the user find collocationally acceptable phrases in written Swedish. As in Futagi et al. (2008), three collocation types[1] are being considered: *verb-**noun*** (direct objects only), *adjective-**noun*** and ***verb**-adverb*. Following Malmgren (2002), a collocation is assumed to contain a *basis* (in boldface above) carrying most of the semantic information of the phrase, and a variable *collocate* which our tool attempts to replace. Other definitions are possible and common, Malmgren (2002) and Nesselhauf (2005) discuss many of them.

[1]Futagi et al. in fact use a fourth pattern, ***noun** of **noun***, giving the example *swarm of bees*, which is often represented using a single word in Swedish (*bisvärm* in this case)

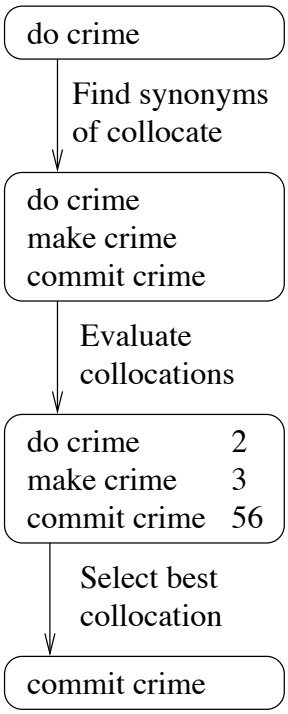

Figure 1: Basic function of the tool

In the phrase *commit crimes* (see also section section 3), *crime* is the basis of the collocation. It carries most of the meaning, and we could easily imagine it being used on its own as a verb. The collocate, *commit*, on the other hand, is rather arbitrary and does not contribute much to the meaning of the phrase. *Do*, *make*, *perform* or *create* might have been used instead, but usually are not.

For a word pair of the three types listed above, the tool tries to find other acceptable collocates. This can be done automatically, using different association measures to test collocation strength. There is also a concordancer, to display examples of usage from the corpus for manual inspection.

brott	Synonymity		Count	MI	LL
	FSL	*RI*			
göra	100%	1.000	8	-1.524	16.88
åstadkomma	80%	0.162	1	0.179	0.015
utföra	92%	0.214	14	1.936	15.21
begå	68%	0.046	439	5.867	2283.0

Table 1: Alternatives to *göra brott*. Synonymity values are from Folkets Synonymlexikon (FSL) or Random Indexing (RI), and are relative to *göra*.
begå brott and *utföra brott* are acceptable in the sense of *commit crimes*.

3 Methods and Algorithms

First, a text corpus[2] is part-of-speech tagged and lemmatized using Granska Tagger (Carlberger and Kann, 1999) and then parsed using MaltParser[3] (Nivre et al., 2007). The result is put in a database, to be used by the concordancer, as well as to obtain the word and word pair statistics necessary for collocation analysis.

The text to be analyzed is similarly tagged and parsed, and the relevant word pairs (collocation candidates) are extracted. A collocation candidate is represented by the lemmas of its components, their part-of-speech tags, and their syntactic relationship.

The variable part of each collocation candidate is then replaced with synonyms from *Folkets synonymlexikon*[4] (Kann and Rosell, 2005) and optionally by similar words according to the Random Indexing model (see section 3.2.2), and the new pairs generated this way are evaluated. This is illustrated in Figure 1.

3.1 Extracting Candidates

After the part-of-speech tagging, lemmatization and parsing steps, each word has been annotated with a POS tag, lemma, head word and relationship to the head word. In the sentence "dogs chase cats" we would, assuming it was correctly annotated, see that the head word of the noun *dogs* (lemma: *dog*) is the verb *chase* and the relationship is *subject*, and that the head word of the noun *cats* (lemma: *cat*) is also *chase*, but here the relationship is *direct object*. *chase* has no head word.

The tool scans through every word in a sentence, and checks if a word and its head word form a pair of any of the types we are interested in. In our "dogs chase cats" example, the only collocation candidate is *chase-cat* (because we are not investigating *subject-verb* pairs like *dog-chase*).

3.2 Evaluating Candidates

In the current system, there are several pieces of information available then evaluating a proposed collocate, listed in Table 1.

3.2.1 Folkets Synonymlexikon

The degree of synonymity with the original collocate is given by the synonym dictionary (Kann and Rosell, 2005). Unfortunately, this is not as helpful as it may sound when dealing with collocations, since the collocate is often used in a specialized or weakened sense. Currently the tool ignores this information by default, but in section 4.2 an experiment is presented that takes the stated level of synonymity into account.

3.2.2 Random Indexing

Random Indexing (Sahlgren, 2005) is an efficient word space model which can be used to obtain a rough measure of the synonymity between words, by measuring the similarity of the contexts they appear in. In this case, the context is limited to the lemmatized basis of word pairs we are investigating in our corpus. Selecting the closest 25 or so words usually gives a list of synonyms (or at least related words), antonyms, and several unrelated words. As we discuss in section 4.2, this list is not very useful on its own, but it can be used to complement Folkets Synonymlexikon, whose main problem is having too few synonyms.

3.2.3 Instances in the Corpus

The number of instances often provides valuable clues, and Park et al. (2008) use this measure as the

[2]The corpus used consists of cleaned-up articles from the Swedish Wikipedia, the Swedish PAROLE corpus and recent Swedish newspaper articles. The subset used for this paper consists of 58,420,604 words, with 5,404,250 (detected) instances of the word pair types under consideration.

[3]`http://maltparser.org/`

[4]`http://lexin.nada.kth.se/synlex`

primary indicator of collocational strength in their tool. In Table 1 we see that it roughly corresponds to the other measures, and to the acceptability of the phrase. However, it can also be misleading. When the collocate is a very common word (such as *göra*), there are usually a few examples due to parsing errors, uncommon usages in the corpus, and so on. In this case, we even have a negative correlation between *göra* and *brott*, and yet there are about as many examples of *göra brott* as of the more acceptable *utföra brott*.

3.2.4 Mutual Information

Mutual Information (Evert, 2005, p. 85) is used to measure the strength of association between words. It is defined by Equation 1, where O_{bc} is the observed frequency of the pair b (basis) and c (collocate), and $E_{bc} = \frac{f_b f_c}{N}$ is the expected frequency, where N is the total number of word pairs of the same type (e.g. *adjective-**noun***), and f_b and f_c are the frequencies of the basis and collocate respectively, in word pairs of this type.

$$\mathrm{MI} = \log \frac{O_{bc}}{E_{bc}} \tag{1}$$

A large positive value (such as for *begå brott* in Table 1) indicates a strong co-occurrence, while a negative value (such as for *göra brott* in Table 1) indicates that the word pair is *less* common than would be expected by chance.

3.2.5 Log-likelihood

The log-likelihood measure (Evert, 2005, p. 83) indicates how confidently we can reject the hypothesis that the distributions of the basis and the collocate in a collocation candidate are independent.

A high value simply indicates that there is some correlation between the words. For instance, in Table 1 we can see that *göra brott* has a rather large LL value because there is a *negative* correlation. From a theoretical point of view this seems like a disaster, but as we will see in section 4, it does not appear to cause too much trouble in practice.

3.2.6 Discussion

Park et al. (2008) observed that collocation improvement tools, in their current error-prone stage of development, should be used with care and a concordancer.

None of the different association measures discussed above produce perfect results. In section 4 we will discuss the practical results, but first we look at some theoretical problems.

Very rare word pairs can have a high Mutual Information value, if the individual words are also rare. This can cause the tool to believe that two words are strongly correlated, even if there is little evidence for this. On the other hand, sometimes this is exactly what we want, since most word pairs are very infrequent. (Evert, 2005, p. 89) discusses various heuristics to work around this issue.

The log-likelihood measure does not distinguish between positive and negative correlation between words. This is, in theory, a big problem since the tool considers any strongly associated word pair a "good" collocation.

In section 4.2 we use the product of the Mutual Information and log-likelihood measures. This product is large and negative for significant negative correlations, close to zero for uncorrelated data or data lacking statistical significance, and a large positive value for significant positive correlations. The practical difference from using only the log-likelihood measure, however, is insignificant.

4 Results

We carried out two experiments to evaluate the accuracy of our tool. In the first experiment, an entire proof-read, professional text was processed, and the suggested changes were evaluated. In the second experiment, a list of non-native-like collocations (popularly known as "collocation errors") was given to the tool, and the suggested alternatives were evaluated.

4.1 Evaluating a Professional Text

Table 2 summarizes the results of the tool on a 5,164 word text using proper Swedish. The tool was made to not change any collocates so that the resulting word pair has less than two occurrences in the corpus. This was done due to the varying quality of the corpus texts, and the fact that the imperfect parsing may result in some words being paired incorrectly, both sources of "incorrect" (or very uncommon) word pairs.

For this experiment, we considered only synonyms given by Folkets Synonymlexikon, and only compared the result when using the highest Mutual Information or highest log-likelihood score to select the collocate.

As we can see, and as we would expect, the vast majority (around 85% to 90%) of collocation can-

	MI	*LL*
Acceptable	46	31
Questionable	(14) 15	(6) 7
Wrong	(19) 23	(12) 14
Total changed	(79) 84	(49) 52
Total unchanged	53	85
Total pairs	535	535

Table 2: Results for a 5,164 word text in good Swedish. The numbers in parenthesis exclude parsing errors.

didates in a professional proof-read text are kept intact. In some of these cases, this is simply because there are no synonyms listed for the collocate (about 25% of cases). Other times, there were less than two instances in the corpus of the pair, which was then ignored (about 50% of cases). In other cases, some alternatives are considered, but rejected (the *total unchanged* row in Table 2).

In some cases, the tool suggests changing the collocate. The results are presented in the first three rows of Table 2. The resulting phrases have been classified (by the authors) as "correct" if the phrase is acceptable and very close in meaning to the original, as "questionable" if there is a slight change in meaning (such as from *large debt* to *enormous debt*) or if we are unsure about the classification, or as "wrong" if there is a considerable change in meaning or if the resulting phrase would not be used.

The total error rate can be obtained by dividing errors (defined here as the *wrong* and *questionable* rows of Table 2) by the total number of word pairs with more than one candidate considered (137). This yields 28% when using Mutual Information and 15% when using log-likelihood.

We could also count only the error rate that the user sees, that is, among the word pairs that are actually changed by the tool. This gives 45% (MI) and 40% (LL).

One should keep in mind that these numbers represent the most difficult cases, where we have several credible candidates (occurring at least twice in the corpus *and* being listed in the synonym dictionary) available to choose from. If we count all 535 word pairs considered, we arrive at error rates of 7.1% (MI) and 3.9% (LL) instead.

	Correct	*Top 3*	*Wrong*
FSL	30	2	27
RI	11	4	44
FSL+RI	34	3	22

Table 3: Different methods of selecting synonyms.

4.2 Finding Acceptable Collocations

A list of 60 non-native-like collocations was compiled by taking existing collocations in Swedish, and replacing the collocate with a synonym or other related word, such as a word which shares the same English translation, creating a phrase which was not acceptable to a native speaker.

30 of the items were based on *verb-**noun*** collocations, and the remaining 30 on *adjective-**noun*** collocations. No ***verb**-adverb* collocations were used, because Folkets Synonymlexikon does not differ between different parts of speech, resulting in poor coverage of adverbs (which are regularly confused with adjectives).

The tool was then asked to find and rank alternatives for the word pairs in the list. We classified the result into the three columns of Table 3: *Correct* (if the top candidate was acceptable and of the intended meaning), *Top 3* (if the top candidate was not acceptable, but at least one among the top 3 candidates was) or *Wrong* (if all of the top 3 candidates were considered unacceptable to a native speaker).

As for the columns of Table 3, *FSL* uses only Folkets Synonymlexikon to suggest synonyms, *RI* uses only the 25 best synonyms to the collocate (including the word itself) according to a Random Indexing measure of similarity, and *FSL+RI* uses the union of these two.

$$s = (FSL + RI) \times MI \times LL \qquad (2)$$

Equation 2 was used to rank the results. *FSL* and *RI* are the Folkets Synonymlexikon and Random Indexing synonymity measures (normalized to the interval $[0, 1]$) between the original and the candidate collocates, while *MI* and *LL* are the Mutual Information and log-likelihood measures of the collocation candidates.

We can see that using both synonym sources, the tool is able to automatically find an acceptable collocation 57% of the times. In another 5% we find at least one good suggestion among the top three. Unfortunately, the method of arbitrar-

ily constructing a test set can give no more than a hint of how useful these algorithms would be in practice. Ideally, we would want to use a large selection of collocation errors from e.g. a learners' corpus, but this would require much more time to obtain and extract.

5 Error Analysis

Each of the components used is a potential source of errors.

The part-of-speech tagger is good but not perfect, Carlberger and Kann (1999) report around 97% accuracy. In our experiments, no errors were found to have been directly caused by the POS tagger.

The parser has a much lower accuracy, and while it is usually sufficient for the relatively simple constructs we are interested in, errors are fairly common. In Table 2 we can see that 5 of 38 bad collocations (in the case of MI; 3 of 21 in the case of LL) were caused by parser errors.

The text corpus used should ideally not contain any errors or very unusual phrases, but this is of course hard to achieve in practice, as is obtaining a large enough corpus. Our corpus of around 60 million words is relatively large by Swedish standards, but can not compare to e.g. the billion-word corpus of Futagi et al. (2008). However, the size of the corpus seems to be sufficient to make it only a minor contributor to the total error rate.

Folkets Synonymlexikon was automatically generated and then improved by anonymous users via the Internet. The result is surprisingly good for many types of words and phrases, but not particularly well-suited for this application. If the word sought after is not listed as a synonym, the tool will not consider it. Using the lemma form of words helps to deal with the many and often irregular inflections in Swedish, but it also makes the tool unable to tell when a certain collocation requires a particular form of a word to be used.

The "synonyms" generated by Random Indexing frequently include antonyms, which is unfortunate since words often co-occur with the same words as their antonyms co-occur with. While association measures like Mutual Information and log-likelihood can usually be used to weed out irrelevant words, antonyms that find their way into the "synonym list" may be ranked very high in the suggestion list.

Finally, the user is likely to cause trouble for the tool, by not writing grammatically correct sentences with perfect spelling. No attempt has been made to handle grammatical or spelling errors.

6 Future Work

Two things in particular can be identified as being both necessary and possible to improve: the corpus and its processing, and the synonym dictionary.

6.1 Corpus

Collecting a good corpus is rarely easy, especially for a language like Swedish with only about ten million speakers worldwide. In addition to general technical and legal problems, our tool requires the corpus to be annotated with part-of-speech and syntactic relations. The programs used for this preprocessing, in particular MaltParser, demand quite a bit of processing power. Hundreds of CPU hours have been spent processing even our modest corpus, whose size is only a few percent of the more than one billion word corpus used by Futagi et al. (2008).

Full parsing of sentences is not necessary, since our task is only to identify word pairs of a few types. Futagi et al. (2008) use regular expressions instead of a full parser, although the reason cited is not performance but the poor results of current parsers on texts with flawed spelling and grammar (which is expected from the non-native speakers their tool is aimed at).

Park et al. (2008) use raw word n-grams. This approach obviously eliminates the processing time, but fails to recognize unusual word orders that are unlikely to occur in the corpus. This is particularly important when the corpus is small, and there are only a few examples of a particular collocation.

6.2 Synonyms

It is clear that for a tool to be useful in practice, a better method of replacing the collocate must be used. Here we will briefly discuss some approaches that have been used or mentioned in the literature.

Chang et al. (2008) show that in the case of Chinese learners of English, the vast majority of collocation errors made share the same Chinese translation, and consequently they design their tool around the idea that the proper word can usually be found by translating the mistaken word liter-

ally to Chinese and back. This approach assumes that a bilingual dictionary is available between the language of the tool and the learner's native language. Unfortunately, this is rarely the case with Swedish.

Malmgren (2002) discusses *lexical functions* from a Swedish perspective, which could be useful for a tool like ours. A common and simple example of a lexical function is **Magn**, which maps the basis (e.g. the noun in an adjective-noun collocation) to a set of possible collocates that "magnify" the basis. For instance: **Magn**(illness) = {severe, serious} and **Magn**(wind) = {strong}. Malmgren (2002) cites studies suggesting there are about 50 to 60 lexical functions in most languages. Given a dictionary of these, one could guess that "strong illness" refers to **Magn**(illness), where the accepted adjectives are *severe* and *serious*.

Shei and Pain (2000) suggest that during classroom use of a "collocational aid," pairs of *actual* and *intended* phrases should be saved. Eventually, most common errors should be covered, and even if the exact phrase is not in the database, one could use it to find sets of words that are often confused.

Acknowledgments

Johan Hall and Joakim Nivre assisted in getting their MaltParser to work with the Granska Tagger. Pontus Stenetorp contributed to the corpus used. The two anonymous reviewers made many useful suggestions.

References

Johan Carlberger and Viggo Kann. 1999. Implementing an efficient part-of-speech tagger. *Software – Practice and Experience*, 29:815–832.

Yu-Chia Chang, Jason S. Chang, Hao-Jan Chen, and Hsien-Chin Liou. 2008. An automatic collocation writing assistant for Taiwanese EFL learners: A case of corpus-based NLP technology. *Computer Assisted Language Learning*, 21:283–299.

Stefan Evert. 2005. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. Ph.D. thesis, Universität Stuttgart.

Yoko Futagi, Paus Deane, Martin Chodorow, and Joel Tetreault. 2008. A computational approach to detecting collocation errors in the writing of non-native speakers of English. *Computer Assisted Language Learning*, 21:353–367.

Viggo Kann and Magnus Rosell. 2005. Free construction of a Swedish dictionary of synonyms. In *NoDaLiDa 2005, Joensuu*.

Sven-Göran Malmgren. 2002. *Begå eller ta självmord? – Om svenska kollokationer och deras förändringsbenägenhet 1800–2000*. Volume 15 of ORDAT. Göteborgs universitet. `http://spraakdata.gu.se/ordat/pdf/ORDAT15.pdf`.

Nadja Nesselhauf. 2005. *Collocations in a Learner Corpus*, volume 14 of *Studies in Corpus Linguistics*. John Benjamins Publishing Company.

J. Nivre, J. Hall, J. Nilsson, A. Chanev, G. Eryigit, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 13(2):95–135.

Taehyun Park, Edward Lank, Pascal Poupart, and Michael Terry. 2008. Is the sky pure today? AwkChecker: an assistive tool for detecting and correcting collocation errors. In *UIST '08: Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 121–130, New York, NY, USA. ACM.

Darren Pearce. 2002. A comparative evaluation of collocation extraction techniques. In *Third International Conference on Language Resources and Evaluation*.

Magnus Sahlgren. 2005. An Introduction to Random Indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering*.

Chris Chi-Chiang Shei and Helen Pain. 2000. An ESL Writer's Collocational Aid. *Computer Assisted Language Learning*, 13(2):167–182.

K – A construction substitute

Gunnar Eriksson
Swedish Institute of Computer Science
Box 1263
SE-164 29 Kista
Sweden
guer@sics.se

Abstract

In this paper a simplistic method of combining structural och lexical information is suggested. The aim is to gain better modelling of some aspects of sentence meaning. The method is easy to incorporate into existing standard word-based approaches, and two example applications are briefly discussed.

1 Constructions and text classification

A construction is usually rather informally defined as a form-meaning pair, that integrate a particular form and a particular meaning in a more or less conventionalized and often non-compositional way. Form can be thought of as any pattern of aggregation: e.g. morphological or syntactic, and meaning may stretch from lexical meaning to discourse.

Underlying this is the position that the meaning of a utterance is not solely to be found in the semantics of the occurring lexical items – the composition of those items in structuring the utterance may also contribute to the full meaning.

On the other hand, a standard text classification task is normally performed by using (a subset of) the containing words as predicting features. Normally this is done by treating the words as elements of an unordered set, and thereby ignoring most inter-dependencies and relations between the containing words. The only remaining relation is their mutual occurrence. This simple approach has proven effective in a large number of different tasks. The experiments presented here raise the question whether it is possible to augment such a standard bag-of-words representation with information about the arrangement of words without complicating the use of the standard model. In (Karlgren et al., 2008) we proposed $Kmarkers$ as such "constructional features". K markers are atomic labels that describes some aspect of the structure in a sentence, and are used, alongside with the words of the sentence, as descriptors of that sentence. As in the case of standard bag-of-words representation, all relations in and between the word set and the K set are ignored. By this expansion of the set of prediction features, the lexical and the structural features are treated the same way, and are given equal importance.

2 K markers

The primary aim of the K markers in the first experiment was to capture aspects of sentence composition, and therefore the majority of markers (cf. Table 1) is concerned with clause types, the way different types of clauses patterns in a sentence, and types of adverbial. All K markers are based on the dependency and morphological analysis of the sentence accomplished by The Conexor Functional Dependency (FDG) parser, (Tapanainen and Järvinen, 1997). Some markers, such as adverbial types and information about predicate and relative clauses, could be picked directly from the FDG dependency or morphology output. Other clause type markers had to be crafted out of information from different levels of the analysis. Examples of such are transitive clauses – including the sub-grouping according to the type of object, intransitive clauses, and the marker for the temporality pattern of clauses. The full list of the 14 K markers of the sentence composition type is, (cf. Table 1): TRIN, TRTR, PREDCLS, TRMIX, TNSSHIFT, OBJCLS, RELCLS, ADVLSPAT, ADVLTIM, ADVLSNT, ADVLMAN, ADVLCOND, ADVLCLSIN, ADVLQUANT.

The next subgroup of the K set are the FDG dependency tags of the sentence that remained after the construction of the first subgroup. The remainder was filtered from technical tags used for punctuation, from tags for internal nominal phrase and prepositional phrase structure, and from tags for

main clauses, subject, and coordination. The exclusion of main clause and coordination tags also excluded the possibility to represent the parataxis-hypotaxis pattern of the sentence in the K set. The tags in Table 1 from the FDG dependency residual group are: SUBCNJ, PPUNDET, NEG, PP-POMOD, VCHAIN, ADJMOD, QUANT, PARTV, OTHER.

The last subgroup of the K set is a couple of markers that are purely morphological. The first subset was selected mainly of technical reasons: since we wanted to investigate the impact of tense patterns, we also needed markers for simple tensed clauses. These are TNSPRES and TNSPAST. The other subset, consisting of markers for the grade of occurring adjectives (ABSADJ, KMPADJ, and SUPADJ), was included because of the intended task: to identify opinionated and attitudinal sentences. We wanted to investigate whether the distribution av different adjective forms could predict the occurrence and polarity of such sentences.

Figure 1 gives some example sentences and Table 1 presents the full set of K traits together with their frequencies in a corpus of approx. 90,000 words in 4,306 sentences of newspaper text.

TENSE SHIFT *It* **is** *this, I* **think***, that commentators* **mean** *mean when they* **say** *glibly That the "world* **changed***" after Sept 11.*

TIME ADVERBIAL *In Bishkek, they agreed to an informal meeting* **later this year***, most likely to be held in Russia.*

OBJECT CLAUSE *China could use the test as a political signal to show the US* **that it is a rising nuclear power** *at this tense moment.*

VERB CHAIN *"Money* **could be earned** *by selling recycled first-run fuel and separated products which retain over 50 per cent of unused uranium," Interfax news agency reported him as saying.*

Figure 1: K examples.

3 Applications of K features

3.1 Identification of attitude

In *the NTCIR-7 Opinion Analysis Task* (Karlgren et al., 2008) we applied the K traits to the task of identifying sentences with expressed opinions and attitudes. The K features were used by classifiers in combination with two other feature sets: I – the set of content words, and F – the set of function words. Two different classifiers were used, a support vector machine and a classifier based on a word-space model. All combinations of the three feature sets were used, and for most of the different experimental settings the classifiers that included the K features performed slightly better in terms of F-score in relation to a manually annotated gold standard corpus. See (Karlgren et al., 2008) for details about performance of the different classifiers and different feature set composition, as well as information about the data sets used for training and evaluation.

When a SVM classifier and a feature set of all three (I, F, and K) feature types was used, 18 K markers were among the top 2245 most predictive features. Table 2 list those markers with their rank and their sub-grouping in the K set and repeats their occurrence in the NTCIR-7 corpus. 85 of the remaining features were function words (F), and the rest, 2142, from the content word I set.

The importance of the K markers compared to the F and I sets in the top 2245 list might be indicated by their joint rank. When the rank-sums of the feature sets are compared by the Mann-Whitney U test, I tends towards the end of the 2245 list ($p > 0.90$), and are somewhat less interesting in the list of features compared to the other features, while K features tend to occur towards the beginning ($p > 0.95$). This suggests high prominence for the top 18 K markers.

The list in Table 2 shows that surprisingly many of the K markers from the morphology and FDG dependency subsets succeeded to defend a position in the top 2245 set. The high position of the NOK trait may also seem odd, but are due to the sentence segmentation of the data set: sequences of extra-textual, "non-sentential" material were assigned sentence status. These text parts were never annotated as opinionated and also got a poor treatment by the FDG analysis. The artificial result of this, is the marker's strong predictiveness for non-opinionated sentences.

Another way to investigate the impact of different K traits is to study their occurrence in the sentences in the NTCIR-7 corpus. A matrix of K features and attitudinal status of sentences was constructed, and reduced to two dimensions by *correspondence analysis*, cf. (Greenacre, 1984). This

Form	K tag	N		Form	K tag	N sent
Non-transitive clause	TRIN	2919		Undetermined prepositional phrase	PPUNDET	22
Transitive clause	TRTR	2350		Negation	NEG	17
Predicative clause	PREDCLS	1439		Prepositional post-modifier	PPPOMOD	572
Transitivity mix	TRMIX	1283		Verb chain	VCHAIN	532
Tense shift	TNSSHIFT	733		Adjective modifier	ADJMOD	82
Object clause	OBJCLS	351		Quantifier	QUANT	69
Subordinating conjunction	SUBCNJ	200		Verb particle	PARTV	7
Relative clause	RELCLS	601		Base form adjective	ABSADJ	3417
Adverbial of location	ADVLSPAT	1367		Present tense	TNSPRES	2373
Adverbial of time	ADVLTIM	1110		Past tense	TNSPAST	2145
Sentence adverbial	ADVLSNT	973		Comparative adjective	KMPADJ	463
Adverbial of manner	ADVLMAN	608		Superlative adjective	SUPADJ	241
Adverbial of condition	ADVLCOND	547		Other	OTHER	5
Clause-initial adverbial	ADVLCLSIN	387		No K-traits	NOK	66
Adverbial of quantity	ADVLQUANT	85				

Table 1: K in NTCIR-7 MOAT corpus.

Rank	K tag	Type	N sent
71	NOK	—	66
75	TNSSHIFT	sentence composition	733
269	TNSPAST	morphology	2145
281	PARTV	FDG dependency	7
290	TRMIX	sentence composition	1283
385	ADVLQUANT	sentence composition	85
502	PREDCLS	sentence composition	1439
505	QUANT	FDG dependency	69
680	TRIN	sentence composition	2919
686	NEG	FDG dependency	17
746	ADVLTIM	sentence composition	1110
780	TRTR	sentence composition	2350
813	PPPOMOD	FDG dependency	572
969	PPUNDET	FDG dependency	22
1055	KMPADJ	morphology	463
1105	VCHAIN	FDG dependency	532
1673	TNSPRES	morphology	2373
2222	ADVLCOND	sentence composition	547

Table 2: K among the topmost 2245 predictive features.

is a method similar to *principal components analysis*, but with the additional feature of placing the column and row variables on the same plane, and thus makes it possible to study the K features occurrence in sentences of different attitudinal type.

Figure 2 shows a plot of the result from the correspondence analysis, with most extreme outliers removed. The opinion analysis tags in the plot are: $YPOS$ = opinionated sentence with positive polarity, $YNEG$ = opinionated sentence with negative polarity, $YNEU$ = opinionated sentence with neutral polarity, and N = sentence without expressed opinion. The proximity of two labels is a measure of their co-occurrence, and we can notice that some K markers predominately show up in non-attitudinal sentences, e.g. verb chains, time adverbials and adverbials of quantity. On the opinionated side of the plane we find K traits as clause objects, tense shift patterns, adjectives in comparative grade, and predicative clauses. We can also see that the opinionated and non-opinionated sentences are spread along the x-axis, the most important of the resulting two correspondence analysis dimensions, while the y-axis seems to involve the polarity of the sentence.

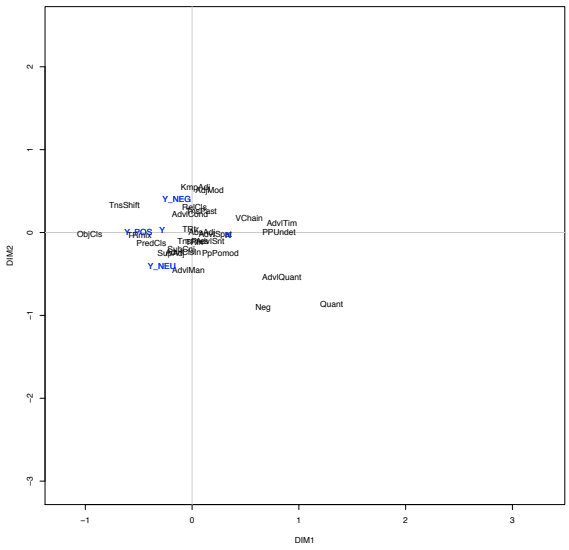

Figure 2: K x Attitude.

3.2 Novelty detection

The impact of K has been further studied in a preliminary experiment. In this, the relation between the K set and the judged novelty of a sentence in relation to a certain topic was investigated. The *TREC 2003 Novelty Track* data set was used, where sentences are assessed relevant (REL), relevant and containing new information (NOV), or irrelevant (BOR). Again, correspondence analy-

sis was used to analyse the data, and the result is plotted in Figure 3. The K set used is the same as the one presented above, although the labels are different. The individual identity of the K markers is irrelevant at this point, however, since the aim of this preliminary study was to investigate how the K set was distributed in the data. The plot shows traces of an uneven distribution: the novelty labels REL, NOV, and BOR are separated by their co-occurrence with K markers, and it seems to be a distributional difference between relevant (REL + NOV) and irrelevant (BOR) sentences, as well as a difference between BOR + NOV versus REL. Note that this experiment does not have anything to say about the overall usefulness of K markers in this kind of task, but only suggests that authors to some extent do use structural means to express the novelty/relevance of a sentence, and that these means may be traceable by the use of K attributes.

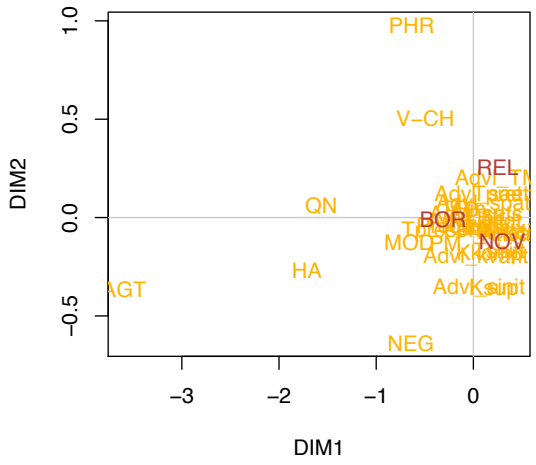

Figure 3: Novelty x K.

4 Discussion

The suggested K features might be one way of mimicking the contribution of constructions to the meaning of a sentence or utterance by combining these atomic markers of structure with the lexical items, without the need to represent the interrelations between the two different sets. But the current implementation of the K traits idea has a number of drawbacks and limitations. The current set of K attributes was selected heuristically with task of attitude identification in mind. A small preliminary experiment may suggest some usefulness even for other tasks, but the generality for wider application is unknown. Data-driven procedures for systematic selection of structural markers are necessary, and would make it possible to build a general palette of possible K candidates.

References

M. J. Greenacre. 1984. *Theory and applications of correspondence analysis*. Academic Press, London.

Jussi Karlgren, Gunnar Eriksson, and Oscar Täckström. 2008. Sics at ntcir-7 moat: constructions represented in parallel with lexical items. In *Proceedings of The 7th NTCIR Workshop (2007/2008) - Evaluation of Information Access Technologies: Information Retrieval, Question Answering and Cross-Lingual Information Access*, page 4, Tokyo, Japan.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *In Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71.