SWORD: Towards Cutting-Edge Swedish Word Processing

Fabienne Cap* **Yvonne Adesam†** **Lars Ahrenberg‡** **Lars Borin†**
Gerlof Bouma† **Markus Forsberg†** **Viggo Kann°**
Robert Östling⋆ **Aaron Smith*** **Mats Wirén⋆** **Joakim Nivre***

*Uppsala University †University of Gothenburg ‡Linköping University °KTH ⋆Stockholm University

Abstract

Despite many years of research on Swedish language technology, there is still no well-documented standard for Swedish word processing covering the whole spectrum from low-level tokenization to morphological analysis and disambiguation. SWORD is a new initiative within the SWE-CLARIN consortium aiming to develop documented standards for Swedish word processing. In this paper, we report on a pilot study of Swedish tokenization, where we compare the output of six different tokenizers on four different text types. For one text type (Wikipedia articles), we also compare to the tokenization produced by six manual annotators.

1. Introduction

When it comes to language technology, Swedish is far from being under-resourced. Over the years, we have seen the development of many annotated corpora, dictionaries and tools for natural language processing. However, because these resources have been developed at different points in time and sometimes for different purposes, they do not conform to a common standard and it can sometimes be hard to know which resources are compatible with each other. Thus, if we want to train a part-of-speech tagger, there is really no alternative to using the Stockholm-Umeå Corpus (SUC) (Ejerhed and Källgren, 1997). If we also want a parser, we have to resort to Talbanken (Einarsson, 1976), but Talbanken does not use the same tokenization and part-of-speech tags as SUC. If we want to throw in a rule-based morphological analyzer, we can use SALDO (Borin et al., 2013) but then we have to face further discrepancies. The development of systems like Granska (Carlberger and Kann, 1999) and STagger (Östling, 2012) have attempted to overcome these problems by harmonizing or modifying resources, which has led to new subtle differences.

A fundamental problem is that we do not even know in most cases how large the discrepancies are or what impact they have on the performance of our systems. This is true in particular of the lowest level of processing, where we break a text into sentences and sentences into words or tokens. The SWORD project is a long-term effort within SWE-CLARIN that aims to develop better documented standards for Swedish word processing, ranging from tokenization to morphological analysis, lemmatization and tagging. In this paper, we report a pilot study on Swedish tokenization, consisting of two experiments.

In the first experiment, we let five project members manually tokenize texts from Wikipedia. The purpose of this experiment is to see whether there is a consensus among scholars in the field about how to tokenize Swedish text but also to establish a standard against which we can compare automatic tokenizers. In the second experiment, we run six existing tokenizers for Swedish on the same Wikipedia texts but also on three different texts taken from newspapers, blogs, and computer manuals, respectively. The purpose of this experiment is to find out what differences exist between different tokenizers and to what extent their output conforms to the standard set by humans. In addition, we hope to get some idea of how sensitive the tokenizers are to different text types.

2. Manual Tokenization

The manual tokenization was performed on extracts from three Wikipedia articles on sewing machines, the second Lord of the Rings movie, and Tyler Oakley, amounting to about 5,000 words in total. Annotators were asked to put each token on its own line and to indicate sentence boundaries with empty lines.

We found that the annotators widely agreed in their tokenization decisions. However, whereas annotators 2–5 all produced a single level of tokenization, annotator 1 chose to devise two levels of segmentation, one low-level mechanical tokenization defined by changes in character class (1a), and one high-level segmentation into lexical units, including multiword expressions (1b). Most of the differences were therefore found between 1a and 2–4 or 2–4 and 1b.

Examples of deviations are given in Table 1. From blocks **A** and **B** it can be seen that 1a and 1b sometimes deviate from all other segmentations. In block **C** we give examples of expressions where more than one annotator deviated from the majority tokenization. Note, however, that all deviations we found may be interpreted as different levels of token granularity, with the single-level tokenization in 2–5 merging low-level tokens in 1a and at the same time providing the building blocks of higer-level units in 1b. All in all, the different tokenizations are therefore highly compatible with each other.

3. Automatic Tokenization

For the automatic tokenization, we extended the set of texts to be processed. In addition to an extended version of the Wikipedia texts, we used newspaper text, computer manuals, and blog texts (about 25,000 words in total).

		Annotator					
		1a	2	3	4	5	1b
A	sick-sacksöm		X	X	X	X	X
	sick - sacksöm	X					
	sick-sacken		X	X	X	X	X
	sick - sacken	X					
	Overlock-symaskiner		X	X	X	X	X
	Overlock - symaskiner	X					
	s.k.		X	X	X	X	X
	s . k .	X					
	enkel-		X	X	X	X	X
	enkel -	X					
B	1773-1857						X
	1773 - 1857	X	X	X	X	X	
	6 000-9 000						X
	6 000 - 9 000	X	X	X	X	X	
	och/eller						X
	och / eller	X	X	X	X	X	
C	Help!		X		X	X	X
	Help !	X		X			
	”the Red S Girl”-trademark						X
	-trademark			X	X	X	
	- trademark	X	X				
	1,5 miljoner						X
	1,5		X	X	X	X	
	1 , 5	X					

Table 1: Manual tokenization differences (sample).

3.1 Tools

We compared six tokenizers:

(1) SwePipe (Swedish Annotation Pipeline) is a tokenizer originally developed in the Swedish Treebank project (Nivre and Megyesi, 2007) and further developed within the Universal Dependencies project (Nivre et al., 2016) to be compatible with the retokenized version of the Swedish Talbanken treebank[1].

(2) SPARV has been developed in conjunction with the Korp project at Språkbanken. Recently, it has become an independent component which can also be used outside of Korp[2]

(3) SSP-SUC is a supervised structured perceptron-based tokenizer trained on SUC[1].

(4) setokenizer is a tokenizer for Swedish text written in Perl and used in the LinES-project (Ahrenberg, 2007). Strings of alphanumeric characters are generally treated as single tokens whereas non-alphanumeric characters are divided into different sets according to their behavior. Users may specify lists of exceptions to adapt the tokenizer to their needs.

(5) Granska is a tokenizer developed at KTH as part of the grammar checker with the same name (Domeij et al., 1999).

(6) UDPipe is a generic trainable pipeline developed for treebanks in Universal Dependencies (Straka et al., 2016). It has been trained on the Swedish UD data without any language-specific enhancements.

[1]https://github.com/robertostling/efselab

[2]https://spraakbanken.gu.se/swe/forskning/infrastruktur/sparv

		Tokenizer					
		1	2	3	4	5	6
A	Ring)	X	X	X		X	X
	Ring)				X		
	smäll !	X	X	X		X	X
	smäll!				X		
	och / eller	X	X	X	X		X
	och/eller					X	
	bland<TOK>annat	X	X	X	X		X
	bland annat					X	
	s . k .						X
	s.k.	X	X	X	X	X	
	EU - nämnden						X
	EU-nämnden	X	X	X	X	X	
B	1900 - talet	X					X
	1900-talet		X	X	X	X	
	1975 / 1976		X				X
	1975/1976	X		X	X	X	
	1773 - 1857	X	X				X
	1773-1857			X	X	X	
	91 : an	X		X		X	X
	91:an		X		X		
	J.R.R. Tolkien			X		X	
	J.R.R . Tolkien	X	X		X		
	J . R . R . Tolkien						X
	osv .		X	X		X	X
	osv.	X			X		
	http :// / www.bernina.se /	X		X			
	http://www.bernina.se/		X		X	X	X

Table 2: Automatic tokenization differences (sample).

	A5	T1	T2	T3	T4	T5	T6
s . k .							X
s.k.	X	X	X	X	X	X	
Help !		X	X	X	X	X	X
Help!	X						
J.R.R. Tolkien				X	X	X	
J.R.R . Tolkien		X	X				
J. R. R. Tolkien	X						
J . R . R . Tolkien							X

Table 3: Manual vs. automatic tokenization (sample).

3.2 Results

Some representative examples taken from the Wikipedia texts are given in Table 2. We see that Granska (5) sometimes deviates from the other tokenizations as it allows for multi-word tokens, like 'bland annat' (= 'among others') which it considers to be one single token in contrast to the other tokenizers. We also see that SwePipe (1) tokenizes numerical expressions inconsistently, treating n-n as one token and n/n as three tokens, and that UDPipe (6) often deviates from the other tokenizers in that it segments into smaller units. Table 3, finally, compares one manual annotator (A5) to all six automatic tokenizers, showing some characteristic differences.

4. Conclusion

This paper takes a first step towards standards for Swedish word processing by comparing the tokenization of manual experts to that of automatic tokenizers. In the future, we will perform a deeper quantitative and qualitative analysis of the results.

5. Acknowledgement

This work has been supported by an infrastructure grant from the Swedish Research Council (SWE-CLARIN, project 821-2013-2003).

References

Lars Ahrenberg. 2007. LinES: An English-Swedish parallel treebank. In *Proceedings of the 16th Nordic Conference of Computational Linguistics (NODALIDA)*, pages 270–273.

Lars Borin, Markus Forsberg, and Lennart Lönngren. 2013. Saldo: A touch of yin to wordnet's yang. *Language Resources and Evaluation*, 47(4):1191–1211, December.

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

Rickard Domeij, Ola Knutsson, Johan Carlberger, and Viggo Kann. 1999. Granska - an efficient hybrid system for swedish grammar checking. In *Proceedings of the 12th Nordic Conference of Computational Linguistics (NODALIDA)*.

Jan Einarsson. 1976. Talbankens skriftspråkskonkordans. Lund University, Department of Scandinavian Languages.

Eva Ejerhed and Gunnel Källgren. 1997. Stockholm Umeå Corpus. Version 1.0. Produced by Department of Linguistics, Umeå University and Department of Linguistics, Stockholm University. ISBN 91-7191-348-3.

Joakim Nivre and Beáta Megyesi. 2007. Bootstrapping a Swedish treeebank using cross-corpus harmonization and annotation projection. In *Proceedings of the 6th International Workshop on Treebanks and Linguistic Theories*, pages 97–102.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*.

Robert Östling. 2012. Stagger. *Northern European Journal of Language Technology*, 5.

Milan Straka, Jan Hajic, and Jana Straková. 2016. UDPipe: Trainable pipeline for processing CoNLL-U files performing tokenization, morphological analysis, POS tagging and parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*.