Experiments on sentence segmentation in Old Swedish editions

Gerlof BOUMA, Yvonne ADESAM

Språkbanken, Department of Swedish
University of Gothenburg

gerlof.bouma@gu.se, yvonne.adesam@gu.se

ABSTRACT

We present experiments on automatic segmentation of electronic Old Swedish editions into sentence-like units. Our target material is characterized by a great variation in the type of boundaries that are marked orthographically, the extent of boundary marking, and the means of boundary marking. We begin with an exploration of boundary marking in a large, unannotated corpus of Old Swedish texts. Then we show that we are able to improve upon a simple but effective segmenting baseline, using a conditional random field model trained on a manually annotated corpus. A more valuable lesson the modelling teaches us, however, is that we need to address the boundary marking variation explicitly.

KEYWORDS: Sentence-like units, boundary detection, Old Swedish.

Gerlof Bouma and Yvonne Adesam 2013. Experiments on sentence segmentation in Old Swedish editions. *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013.* NEALT Proceedings Series 18 / Linköping Electronic Conference Proceedings 87: 11–26.

1 Introduction

Historical text corpora have recently gained much interest in computational linguistics. Due to differences between the historical and contemporary materials, it is not always the case that we can reuse automatic methods developed for contemporary language on the historical texts. In our efforts to compile and process a corpus of Old Swedish texts (13–16th c.), we have found that tools for modern Swedish cannot be effectively applied. We are facing a situation similar to treating a whole new language. One of the major challenges in this material is the lack of a single orthographic standard. This results, for instance, in a wide variety of spellings – the same word may even be spelled in different ways in the same paragraph. Another consequence is a variety of boundary marking strategies, so that it is hard to determine where one sentence ends and another begins. The methods for marking sentence-like units range from a period followed by a word starting with an uppercase letter, over using slashes, commas, or uppercase letters alone, to no marking at all. Sentence segmentation can be helpful to corpus users exploring the texts and is sometimes necessary for computational reasons.

In this paper, we will give a high-level overview of the different boundary marking strategies in our collection (Section 2) and present machine learning experiments on the sentence boundary detection task in Old Swedish on a hand-annotated corpus (Section 3). In the remainder of the current section, we will motivate our interest in the sentence boundary detection task and discuss some relevant previous work.

1.1 Motivation

Segmentation of a text into tokens and sentences is the first step in the traditional NLP pipeline model, minimally consisting of segmentation followed by part-of-speech tagging followed by parsing. The interesting and hard problems occur further down in the pipeline, whereas the problem-free execution of the first step is typically taken for granted. Irrespective of whether sentence segmentation really is a 'solved problem' in modern, professionally written and edited English, it is certainly not the case for our Old Swedish material. Currently no segmenter exists and, as we hope to make clear in the rest of the paper, no single strategy exists that could give good enough results to feed the NLP pipeline. As many annotation tools at later stages rely on having bounded strings to operate on, the pipeline model breaks down without a way to segment the data early on.

Given the lack of clear boundary marking strategies in the material, we might consider deviating from the traditional NLP pipeline model. If we have access to, for instance, a parser that can handle unbounded strings, we could use this to annotate the stream of tokens directly and skip the lower-level segmenting task. If a graphic sentence-like segmentation of the text is desired for other reasons, one could then reconstruct this from one of the other levels of annotation. For instance, the maximal syntactic unit, the *macrosyntagm* (Loman and Jørgensen, 1971), could serve as a suitable unit.

Although we certainly hope to investigate this option in the future, the investigation and experiments presented in this paper are still valuable for a number of reasons. Even if we would have the type of part-of-speech taggers and/or parsers available that do not need pre-segmented data, these tools will probably benefit from information about boundary marking present in the input. This information could come from a stand-alone model, whose predictions are taken into account by the higher level annotation tool. Alternatively, the higher level tool will need to learn about boundary marking itself, in which case we need to supply it with knowledge about

the possibilities and their distribution. In either case, the research presented in this paper will be of help.

Finally, there is also a non-computational reason for the work in this paper. Segmentation is very helpful for corpus users. Searching an unsegmented text and inspecting hits in an unbounded string is cognitively straining. Put concretely, one doesn't know where to start reading. Higher level tools, like a parser, that could give us a useful segmentation as a byproduct have not yet been developed for Old Swedish, nor do we have the necessary annotated data available. The data and dedicated segmentation models investigated in this paper could immediately help to improve the presentational situation.

1.2 Background

Following Stevenson and Gaizauskas (2000), we distinguish between punctuation disambiguation and sentence boundary detection. Segmenting a text into sentences or sentence-like units sometimes reduces to the much more restricted and arguably easier task of punctuation disambiguation. The task is then, given a number of target delimiters like full stop, exclamation mark, question mark, to decide whether they end a sentence or not. In the case of the full stop in English orthography, this means for instance choosing between a sentence final full stop and an abbreviation final period. This restricted task can be solved with near perfection for well-behaved corpora. Reported error rates are well below .5% on professionally written data (Mikheev, 2002; Gillick, 2009). Even completely unsupervised systems are able to achieve error rates below 2% (Kiss and Strunk, 2006) in this task on the same data (see Gillick, 2009, for a comparison). The idea that punctuation disambiguation error rates give a good idea of the performance of the reported systems as segmenters has been challenged, though. See Read et al. (2012) for discussion and a more realistic assessment of the state-of-the-art.

Sentence boundary detection is a more general and harder task in which a language stream (text, transcribed speech) has to be segmented into sentence-like units on the basis of a mixture of information sources, without being able to reduce the segmentation into the recognition or disambiguation of a single signal. This problem description applies in the case of speech data, where the task is often formulated as a labelling problem. Each token in the speech stream is either a boundary token or a non-boundary token. The models for this decision use different types of information, like n-gram language model probabilities and prosodic features taken from the acoustic signal (Gotoh and Renals, 2000; Liu et al., 2005). The latter report a token-based error rate of just below 3.5% for detecting boundary tokens in broadcast news data.[1]

For our Old Swedish material we are dealing with the sentence boundary detection task, because writers mark a variety of different types of boundaries besides sentence-like units, with a variety of marking strategies. In a sense, punctuation symbols and capitalization are to us what prosodic features are to the speech segmentation researcher: an important but uncertain source of information to supplement the lexical level. There may be overall tendencies in the data, but the observed variation makes these non-obvious. Variation is seen looking across documents – which span a time frame of about 3 centuries, are produced at many different places and, not unimportantly, have passed through the hands of several editors – but also within them. Consider the following four sentences. For the larger part, the writer of the document uses a

[1] Note that there is no way of directly comparing the error rates between the two tasks, although it might be possible to convert the punctuation disambiguation error rate by taking the rate of occurrence of disambiguation points into account. Also recall that error rates need to be seen in relation to their majority class baselines.

slash '/' to mark (presumably) some kind of prosodic phrasal break. However, in the last part of the document, no marking whatsoever is used. The sentences below are taken from the part in the document where this strategy switch occurrs. (Henceforth, '‖' marks sentence boundaries inserted as annotation after the fact.)

(1) sua ma han þem æpte sinum vilia bøgha / at þe mago kallas oc vara / þe hælgo kirkiu dorakroka / ‖ mz þe hælgho kirkio gulfe / menas biskopa / oc værulsleke klærka / ‖ þera giri ær sua diup / þæt þær ma ingte i grynna / oc þera høghfærfe oc skøro lifærne gar af fragh [...] ‖ þætta ma pafin an han vil mykyt at bættra ‖ raþe allum biskopum þy følghia i gozs oc allum andrum þingum sum þu hørþe at honum var raþet siælfum at gørra
'Thus he may bend them after his will, so that they may [truly] be called the holy church's door hinges. With the holy church's floor, the bishops are meant and secular clergy. Their greed is so deep that nothing can reach its bottom, and their vanity and sinfulness is infamous [...]. This the pope can make a lot better, if he wants. Advise all bishops to do in questions of property and all other things like you heard he himself was advised to do.' (*Brigitta-autograferna*).

Other prominent marking strategies involve full stops, commas, colons, semi-colons, capitalization and combinations of a delimiter and capitalization. Even if our task reduces to punctuation disambiguation for a given part of a given document, this is by no means the case in general.

To our knowledge, historical sentence boundary detection is not a well researched area. Previous work has followed the speech processing literature by treating the problem as a labelling task. (Huang et al., 2010) segment Classical Chinese and 19th century Chinese, material which lacks any marking so that all information has to come from the lexical text level. They report f-scores of around .84 for their data set. (Petran, 2012) reports experiments on an artificial historical data set, created by removing capitalization and punctuation marks from a modern German newspaper corpus. Using part-of-speech information, he is able to reconstruct sentence breaks with an f-score of .65 and without part-of-speech tags with an f-score of .50. He also shows that reconstructing smaller units like (syntactic) clause and NP/PP chunk is an easier task which may be worthwhile to use as an intermediate step.

We will follow this research and treat segmentation like a labelling task. Before we present our experiments, we will give an overview of different boundary marking strategies in our unannotated corpus.

2 Exploring Sentence Boundaries at Text Level

Our unannotated data – the corpus we eventually wish to make available with annotations – has been supplied by Fornsvenska Textbanken[2] and consists of close to 3 million words in about 150 texts. The corpus contains fiction, legal and religious prose, with texts from the 13th to 16th century and ranging from 200 to 200 000 tokens in length. We have excluded verse from our investigations, because it comes with its own set of conventions.

To get an overview of various boundary marking clues in the texts, we start by extracting non-alphabetic characters, such as *: ; , /* . · – and ¶. We then calculate token-delimiter ratios for each text, that is, the frequency of occurrences of a single delimiter, expressed as the average

[2] `http://project2.sol.lu.se/fornsvenska/`

Marker:	.	.+Cap	,	,+Cap	/	/+Cap	Cap
T-d ratio:	5.22	14.20	86.74	86.90	86.90	86.90	12.82

Nv ærum wi skyldughi at tiunda af alle sæþ ware. || Tiunþ scal i akrum af tæliæs || byriæ at þem skyli fyrstum vp skars oc af bars oc tæliæ swa sum til tiu. || byriæ owan at akri oc lyctæ wiþ ændæ. || byriæ ater þær wiþ ændæ tæliæ. oc lyctæ owan warþa. || fari swa æ mæþæn akra winnæs. || ei scal korn mællum akra bæræ. || Tiund þæsse scal i þry skiptes || taki prester en skyl. || twa före bonde hem til sijn. oc i þry skipti. en lot kirkiuni annæn biscupi. þriþiæ fatökum mannum. || bönder aghu presti til tiunþ sinnæ sighiæ þa þe laþa wilia vm þrea sunnudagha oc lagha wærn vm halda. || Warþar hon ei gömþ innæn þe þriæ sunnudagha. oc warþer hon ætin eller spilt. böte þen ater tiunþ sum þet vlti at prester scaþa fik. || Hawi oc prester siælwer scaþa æn han ei gömer innæn þe þriæ sunnudagha. oc ærum wi skyldughir at tiunda af alle þe sæþ sum man arwþer i iorþena.

Figure 1: Token-delimiter ratios for *Södermannalagen* with an example fragment.

number of tokens between them. We also consider capitalization as a boundary marking strategy, and calculate corresponding token-delimiter ratios for capitalization alone and combinations of any of the punctuation symbols and a following capital.

We hope the token-delimiter ratios can give us clues about the type of segments they delimit. We use this with caution, however, since a single text may use more than one kind of delimiter or not mark all segment boundaries; in both cases, segments are actually shorter than the token-delimiter ratio indicates.

Figure 1 shows an example text and token-delimiter ratios for the most common markers. In this text, capital and period followed by a capital have token-delimiter ratios in the 10–30 range, a range we consider to hint at delimiters that are used to mark sentence-like units. Looking just at the surface features of the example text, we see that in this fragment, capitals reliably indicate sentence boundaries. Periods alone, however, mark a much smaller unit. This is reflected in the low token-delimiter ratio for periods.

Of the 149 texts, about three quarters contain one or several delimiters with a token-delimiter ratio in the 10–30 range. From this way of looking at the data, capitalization appears to be the most prominent single clue, falling in the 10–30 range for 94 documents and rarely exceeding 50. In the machine learning experiments of the next section, we will also see that capitalization is an important clue, although it is neither universal nor perfectly reliable.

In Figure 2 is a fragment from a text where none of the studies delimiters have a token-delimiter ratio below 70. The most frequent markers are comma and capitalization, although the high token-delimiter ratios suggest that many of the sentence boundaries will simply lack surface marking. This is illustrated in the fragment in the same figure. Although some of the sentences end in a comma, most of them have no boundary marker. This clearly demonstrates that segmentation for this material does not reduce to punctuation disambiguation.

If we order the texts after their approximate production period, we notice an interesting trend. Texts where commas have token-delimiter ratios in the 10–30 range are more common in the later periods. This fits nicely with claims made in Svensson (1974), who names the comma as the most common delimiter – marking 'longer pauses' – at the end of the Old Swedish period. Svensson also characterizes the slash '/' as a marker for pauses. We do note, however, that 16

Marker:	.	.+Cap	,	,+Cap	/	/+Cap	Cap
T-d ratio:	435.80	435.80	72.63	326.85	435.80	435.80	77.82

Thaa kesarinnan fik höra ath then wnga herrän war kommen thaa tilreddhe hon sigh mz iomfrum och klädom som hon aldra bäst kunne och kom gaangande til konungen och tilhans son ther the saatho baaden til samman || konungen bad hänne sätia sig när sonnen || kesarinnan sade til konungen herra är tättha edher son som saa länghe hafuer borto waridh när the wisa mästara || konungen sade ya män jak kan ekke wettha hurw thz gaar til thy han wil inthe tala, || thaa sadhe hon herra antuarden honnom mik jak skal wil wäl göra honnom talande och togh honom widh haandena och wille hafuan mz sigh, || thaa warde han sig och wille ekke mz, || fadren bad honom gaa mz hänne || thaa negh han sinom fader ödhmyuklighan och war honom lydoger || kesarinnan ledde honom in j en kammara och badh alla wtgaa och satte honnom oppaa en sänga stok när sigh och sadhe, hiärtans käre diocleciane, huad stor aastundan och länktan jak hafuer äpter tik haft, fran then första dagh

Figure 2: Token-delimiter ratios for *Sju vise mästare C*, with an example fragment.

texts have a token-delimiter ratio for '/' below 30, of which only 2 are below 10. Its alleged status as a pause marker is therefore not obvious from the data, as we would expect to see a larger proportion of low token-delimiter ratios.

Finally, larger units of texts are often marked with the paragraph sign '¶'. It occurs in more than half of the texts. In half of these cases, the token-delimiter ratio falls between 50 and 500, and in a quarter the token-delimiter ratio is above 1000. The wide range of ratios suggests that the paragraph sign may delimit larger units of different types.

A special way of marking the end of a larger unit is found in *Peder Månsson's Bondakonst*, which uses the combination ':-'.

(2) Ta oxana haffwa draghit oc lösas gnides oc strykes wäl halsen oc ryggen theras mädh handommen, oc wpdrage alla stadz wäl skinneth fran ryggenom || ey lantandis thet lodha widher ryggen thy then sywkkdommen är them ganskans ondher som kallas hwdbända || oc ärw the mykith hethe tha latis ey til ath ätha för än the wenda ather ath flämtha oc swetten är bortagangin, || oc giffwes them litith j sänder äta oc swa meer oc meer, || sidan gifwes them drikka, oc swa giffwes them nogh ätha oc rökthes wäll:-
(*Peder Månssons Bondakonst*)

3 Segmentation as tagging

In addition to the unannotated corpus used in the previous section, we also have access to a smaller, hand-annotated corpus, which we use to train a statistical segmentation model. We treat the sentence boundary detection task as a sequence labelling task, assigning to each token a label that indicates whether it starts a sentence or not.[3]

[3]As pointed out by an anonymous reviewer, one can also project sentence boundaries for parallel data such as bible texts, to supplement the manually annotated training data (see, e.g., Haug et al., 2009). This should be explored in future work.

Following (Huang et al., 2010; Petran, 2012) we distinguish five tags in our labelling task. These five tags are:

S: 'singleton', one-token sentence;
L_0, L_1: 'left', first and second token in a sentence, respectively;
M: 'mid', tokens in the middle of the sentence;
R: 'right', last token in a sentence.

Each sentence will be labelled with a tag sequence conforming to $S|L_0(L_1M^*)?R$. Compared to a minimal tag set, with only the distinction sentence beginning vs everything else, the extra state information improves segmenting results (Huang et al., 2010).[4] When applying the resulting labellers to the segmenting task, a new segment should be started for each occurrence of S and of L_0.

3.1 Data preparation

We construct training data from the HaCOSSA corpus (Höder, 2011), a corpus of Old Swedish texts with partial syntactic annotation, consisting of 13 documents ranging between 500 and 32k tokens. HaCOSSA contains among other things annotation of main/subordinate clauses, which we use to define sentence-like segments. Each left edge of a main clause starts a new segment, as do ¶ marks and text structure elements such as titles and paragraphs. In HaCOSSA, not all subordinate clauses are attached to a main clause. Our conversion of clause annotation to sentence-like structuring in effect integrates all unattached subordinate clause into the first reachable main clause to their left. The resulting corpus has just over 8k sentences with about 137k tokens including punctuation (16.5 tokens/sentence).

The documents in the corpus come from different times, different writing schools and different editions, which means that there is considerable variation in orthography. Variation is also found within documents. The wealth of boundary marking strategies is of course the main focus of this paper, but the variation in spelling is an unwanted source of trouble in our machine learning experiments. Its effect is reduced generalizability. First and foremost, spelling variation will lead to data sparseness. Even if a certain word is a good clue for sentence boundary marking throughout the corpus, it might be that the machine learning fails to pick up on this, because the word is spelled in many ways. Secondly, if the model associates certain spellings of a word with boundary marking, this may create problems for evaluation through cross-validation. Using spelling as an intermediary, the model effectively learns from which document and boundary marking convention a sequence comes, which leads to cross-validation giving an inflated idea of the models performance on new data.

As a step towards addressing these problems, we apply a very crude spelling normalization to the texts. The normalization uses simple character mappings, derived from corpus inspection. They are listed in Table 1. The mappings reduce the type count from just under 21k to just over 16k. For future work, we intend to investigate the application of more refined spelling variation handling techniques (Adesam et al., 2012) to the training material.

[4] Even though it is possible that the CRF tagger assigns a nonsensical tag sequence, such as R L_1, this seldom occurs in our experiments. Petran (2012) makes the same observation. Moreover, nonsensical tag sequences do not affect our ability to segment text on the L_0 tags. Technically, it should be possible to restrict the CRF tagger to only yield legal sequences, but this was not explored in the experiments for this paper.

Raw	Norm	Raw	Norm	Raw	Norm
aa	a	gh	g	yy	y
dh, þ	d	ii, ij, j, jj	i	ø, øø, öö	ö
c, ch, q	k	oo	o	æ, ææ, ää	ä
ee	e	th	t	åå	å
ff	f	u, uu, vv, w, ww	v		

Table 1: Spelling simplification mappings used in the normalized data experiments.

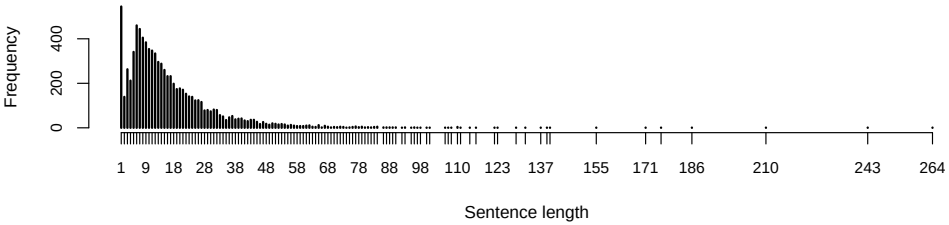

Figure 3: Histogram of sentence lengths in the annotated corpus

3.2 A first look at the data

Before we go into the experiments themselves, let us take a closer look at the data. A histogram of sentence lengths is given in Figure 3. As mentioned above, the mean lies at 16–17 tokens per sentence, and as can be seen from the histogram, the modal sentence has 6–7 tokens. The peak at sentence length 1 consists mainly of ¶-marks, which always constitute their own segment. These are thus trivial to annotate with the S tag.[5] The slight peak at 3 is due to segments of the form ·*XLII*·,[6] which appear in just one of the documents in the set. We note however, that the notation ·NUMERAL· and ·*i*· (either roman numeral 1 or the preposition 'in') is common even sentence internally. Although we can expect a right-tailed distribution of sentence lengths, the extremely long sentences are likely to be an artifact of the way we handled unattached subordinate clauses.

Table 2 gives an overview of the association between lexical items and the L_0, L_1 and R tags, both in terms of the conditional probability of the tag given a token and of the token given the tag. The top ten item lists for each tag were created by sorting the tokens after the product of these two conditional probabilities.[7]

As we can see in the table, the start of a sentence (L_0) is associated with a mix of discourse particles and (predominantly personal) pronouns. The particle *ok* 'and/too' is both very frequent as a first token in a sentence and fairly reliable as a beginning of sentence clue. A more reliable

[5] In the evaluation in the next section, we only look at the accuracy of recognizing the start of a new multi-token sentence L_0. The fact that ¶ is easily recognizable as an S and the preceding token as an R does therefore not skew the evaluation results compared to incorporating ¶-marks into a previous or following segment.

[6] Depending on the text and edition, these dots may also rest on the baseline like (modern) full stops. We show the centred dot examples to set them apart from modern punctuation in the body text.

[7] This product itself is not shown in the tables, as it is just a means to select examples and not a quantity of interest in this discussion. Sorting tokens by the product of p(Tag|Token) and p(Token|Tag) gives the same ordering as the association measures PMI^2 and *geometric mean*, known from the collocation extraction literature (Evert, 2005).

w	$\sum w$	$p(L_0\|w)$	$p(w\|L_0)$	w	$\sum w$	$p(L_1\|w)$	$p(w\|L_1)$	w	$\sum w$	$p(R\|w)$	$p(w\|R)$
ok 'and/too'	9002	.24	.28	*är* 'is'	1516	.22	.04	/	4341	.44	.25
nv 'now'	603	.63	.05	*skal* 'shall'	529	.26	.02	.	4647	.35	.21
ta 'then'	1023	.28	.04	*skalt* 'shall'	153	.48	.01	*svarade* 'replied'	134	.51	.01
sidan 'since'	286	.45	.02	*talar* 'speaks'	125	.42	.01	*sigiande* 'saying'	59	.56	.00
o 'o' (VOC)	132	.58	.01	*äpter* 'after'	183	.33	.01	*etketera* 'etc.'	13	1.0	.00
iak 'I'	967	.20	.02	*sagde* 'said'	272	.27	.01	*ketera* '[et]c.'	13	.92	.00
ty 'for/it'	1171	.15	.02	*vil* 'wants'	163	.33	.01	*sagde* 'said'	272	.18	.01
ter 'there'	663	.19	.02	*ty* 'for/it'	1171	.12	.02	,	307	.17	.01
han 'he/him'	2354	.10	.03	*lifde* 'lived'	55	.53	.00	*sigiandis* 'saying'	15	.73	.00
tz 'it'	820	.16	.02	*man* 'person'	332	.21	.01	*döttir* 'daughter'	17	.65	.00

Note: For reasons of space, morphological features are not given explicitly and only approximated in the English translations. Spelling has been normalized according to the rules in Table 1.

Table 2: Tag-token associations for the L_0, L_1, and R tags in the annotated corpus.

clue, but one that is genre restricted,[8] is the word *nv* 'now', which is used to start a new case in a legal text. A common continuation after such a case starting sentence, is a sentence that starts with *ta* 'then', which introduces the consequences and rules applicable in the case. This combination is illustrated in the following example:

(3) **Nu** hittir man fynd innæn allmannæ leþ. hwat fynd þæt hælzt ær. || **þa** aghi þær aff hwarn attundæ pænning.
'Now someone finds something on a public road, whatever the found thing is. Then [they] have a right to an eighth of its value.' (*Upplandslagen*)

Looking at tokens that associate with L_1, we see mainly finite verbs, a reflex of the verb-second tendency in Old Swedish. The presence of *äpter* 'after' shows a limitation of defining tokens as graphical words, as almost all these cases are part of the discourse connective *ter äpter* 'thereafter'.

In the rightmost table, tokens that hint at the end of a sentence (R), are either punctuation marks or *verba dicendi*. The latter is a result from the HaCOSSA annotation guidelines, where direct speech is seperated from the speech verb (Höder, 2011, s3.6). Note that *etketera* 'etcetera' and its graphically split variant *et ketera* almost exclusively appear at the end of a sentence.

In the previous section, token-delimiter ratios suggested capitalization is a promising candidate for a general segmentation clue. Indeed, in the annotated dataset, capitalization is strongly associated with the L_0 tag. Of all capitalized words, 67% start a new sentence and, vice versa, of all sentences, 57% start with a capital.

3.3 Machine learning experiments

The data inspection of the previous subsection gives us an idea of the kind of features that could be useful when training a statistical labeller. It also gives us an impression of the kind of performance that we should minimally expect from a system. We can read the conditional

[8] The only text of this kind in the HaCOSSA corpus is an edition of *Upplandslagen*. However, the same pattern can be observed in some of the other old laws, too.

Feature	Description
1	current token lower case string
2	Is the current token capitalized?
3	bag of two preceding tokens lower case strings
4	following token lower case string
5	Is the following token capitalized?
6	two character suffix of the current token
7	current token category (numberlike, punctuationlike, wordlike)
1×2	current token × is the current token capitalized?
2×3	preceding token × is the current token capitalized?
3×7	preceding token × current token category
4×7	following token × current token category
1×2×3	current token string × preceding token string × is the current token capitalized?
M	preceding × current tag (first-order Markov assumption)

Table 3: Feature descriptions for the {L_0,L_1,M,R,S} labelling task.

probabilities in the tables and text above as precision and recall. So, a classifier that would assign L_0 to all and only occurrences of the token *ok* would have a precision of .24 and a recall of .28. More interestingly, the classifier that tags all and only capitalized words as L_0 would have a precision of .675 and a recall of .573, giving an f-score of .620. This latter classifier will serve as a baseline in the experiments below.

The association between tokens and certain tags strongly suggests that lexical information should be included as a feature in our tagger. The lexical level is also the only information source that can help us tackle the examples of sentences that lack any surface marking, for which we saw examples in the previous section. Capitalization information is also included as a separate feature. The observation that L_1 associates with finite verbs is hard to model in the absence of some kind of part-of-speech or morphological tagging. However, as a very rough approximation, we include token suffixes into the model: the last 2 characters of each token are used as a feature. In our discussion of sentence lengths, we noted that numerals in the text were often accompanied by special punctuation. To help the model recognize these cases, we include a category feature that divides tokens into one of three categories: *numberlike* (roman numerals or indistinguishable from such), *punctuationlike* (all non-alphabetic characters) and *wordlike* (everything else).

Looking at the results above and at the previous section, it is clear that the model should also be able to use combined strategies, such as: assign R to every punctuation mark that is followed by a capitalized word. To some extent, such tendencies may be captured by making a tag's likelihood depend on the previous tag, but more fine-grained information is possible using conjoined features.

We trained a linear conditional random field tagger with the features listed in Table 3.[9] We evaluate the tagger using cross-validation with two different regimes. First, we perform random

[9]To construct the tagger we used the CRF++ package (`http://chasen.org/~taku/software/CRF++`), with smoothing factor c=0.1 and a feature count threshold f=3.

		Random 10-fold crossval			Leave 1 text out		
Data	Model	P	R	F_1	P	R	F_1
normalized	all features	82.9	66.4	73.7	76.0	58.2	65.9
	no capitalization	79.0	52.0	62.7	71.6	39.3	50.7
	no categories	82.7	66.0	73.4	76.1	58.3	66.0
	no context	77.6	57.7	66.2	73.4	52.7	61.4
	0th order	80.8	62.2	70.3	75.7	55.4	64.0
raw	all features	82.2	65.1	72.7	70.6	58.3	63.9
smurfed	all features	71.2	57.2	63.4	70.3	55.9	62.3
baseline (capitalization)					67.5	57.3	62.0

Table 4: Sentence segmenting model comparison (micro-averaged L_0 tag precision, recall and f-score, per evaluation regime).

tenfold cross-validation. Each paragraph is randomly assigned to a fold, giving 10 more or less equally sized folds. However, since the boundary marking strategies are likely to differ strongly from text to text, a leave-one-document-out evaluation regime will give a more realistic estimate of our segmenter's performance. Note, however, that since document sizes span 2 orders of magnitude, the resulting 13 folds are very uneven in their training/testing data set size ratios.

By comparing results between regimes, we can draw conclusions about how general the models are. If adding or removing features affects the results in the same way in both regimes, this means that the model is not sensitive to properties particular to one document, that is, the model generalizes well over document types. Conversely, if we observe an effect in one regime but not the other, this difference points to a lack of generalizability of the features.

Table 4 shows a comparison of models with access to different kinds of information under the two evaluation regimes. Looking first at the difference between the two regimes, we conclude that random cross-validation gives considerably higher averages than document-based cross-validation – a clear sign that overall the models are struggling to find good generalizations that hold across texts. The spelling normalization, rudimentary as it may be, does help in this respect. If we compare the all-features model trained on original (raw) text to the one trained on normalized text, we see an improvement of more than 5 points in precision in the per-document evaluation regime for the latter.

Focussing on the normalized data and the document-based validation,[10] we see that the capitalization information is the most informative feature, followed by the context information, that is, information about preceding and following tokens. The categorization information does not seem to add anything to the model that can be generalized across documents. The all-features model outperforms the capitalization baseline with almost 4 points in f-score, mostly due to a greatly increased precision.

To see how much the models benefit from the lexical level, that is, the words themselves and not

[10]Note that although the averages in the random cross-validation are much higher in this group of experiments, the trends are very similar.

				Precision		Recall		F_1-score	
	$\sum word$	$\sum sent$	$\sum par$	Bl	Af	Bl	Af	Bl	Af
1	1 074	55	1	-	▪42.9	0.0	▪ 5.5	-	▪ 9.7
2	20 241	1 083	67	68.4	▪84.7	64.9▪	54.3	66.6▪	66.2
3	14 482	631	50	55.5	▪78.5	63.5	▪67.0	59.3	▪72.3
4	3 488	107	19	47.7	▪75.7	76.6▪	72.9	58.8	▪74.3
5	563	38	2	75.0	▪87.2	63.2▪	34.2	68.6▪	49.1
6	1 447	115	14	76.6	▪87.2	42.6▪	29.6	54.7▪	44.2
7	7 439	385	39	48.7	▪68.7	82.6	▪83.4	61.3	▪75.4
8	35 288	2 069	52	91.6▪	86.9	45.6	▪53.7	60.9	▪66.4
9	9 972	345	24	60.1	▪77.2	60.3▪	59.7	60.2	▪67.3
10	22 376	1 566	74	73.8	▪75.0	77.7▪	72.7	75.7▪	73.8
11	18 497	1 229	291	79.6▪	58.7	32.0	▪42.4	45.6	▪49.2
12	3 223	114	11	37.7	▪65.7	78.1▪	57.0	50.9	▪61.0
13	3 255	64	11	22.8	▪63.5	71.9▪	62.5	34.6	▪63.0
			Micro average	67.5	▪76.0	57.3	▪58.2	62.0	▪65.9
			Macro average	61.5	▪73.2	58.4▪	53.5	59.8	▪61.8

Table 5: Per document comparison of the baseline (Bl) and the model including all features (Af). The top results per document are highlighted.

just surface clues like punctuation and capitalization, we also trained models on delexicalized (or: 'smurfed') data. All non-punctuation tokens are replaced by one and the same word, but capitalization is kept as in the original. The loss in both precision and recall shows that the models can successfully use and generalize lexical information. However, the fact that the two evaluation regimes now lead to much more similar results than in any of the other data/model combinations suggests that, even though the lexical information is potentially very useful, it is also particularly hard to generalize.

In terms of error rate and NIST score,[11] the all-features model in the leave-one-document-out regime has an error rate of .034 and a NIST score of .602. The capitalization baseline has .040 and .703 respectively. Labelling none of the tokens as L_0 gives an error rate of .057.

The advantages of using the statistical model over the simple baseline may seem modest, given the amount of information that goes into them. However, even though the capitalization baseline gives a good overall performance, it may be that there are large differences on a per-document basis. By combining different strategies, the statistical model could in principle guard us against the variability between documents. Table 5 breaks down performance per fold in the leave-one-document-out evaluation regime, and gives macro averages in addition to the micro averages used thus far.[12] The table also gives the size of the left out documents, to give an idea of the influence of training set size on performance – when the left out document is up

[11] On the familiar true/false positive/negative contingency table, these are defined as follows: Error rate is $(\textit{false positives} + \textit{false negatives})/N$; NIST score, $(\textit{false positives} + \textit{false negatives})/(\textit{true positives} + \textit{false negatives})$ (Liu and Shriberg, 2007). Note that the NIST score is 1.0 if we label everything as negative and may be above 1.0.

[12] Macro averages: an unweighted average over the precision and recall scores of the folds. Micro average: an average where each fold is weighted by the number of tokens in the test document.

to a fourth of the total data size, one might expect to notice an effect of the reduced training set size. However, no such correlation is visible in the table, suggesting that even if such effects may exist, they are dwarfed by the between-document variability.

The baseline has better macro-averaged recall, but in the other averages, the statistical model outperforms the baseline. Looking at the folds, we can see that the statistical model generally is more precise, whereas the baseline has a higher recall in over half of the folds. Although the model has learned to include other clues, it has also become more conservative in labelling tokens as L_o. The better micro-averaged recall of the statistical model is due to its better recall on some of the larger folds, like 3, 8 and 11.

The advantage of having other clues does show in fold 1, which does not use any capitals. The text uses '/' to mark a smaller unit (token-delimiter ratio: 8.4).

(4) enne persona syntis vakande / oc eg sofande / sum hon vare i eno palacio / || oc i fræste væginne syntis en sol / myok stor || framan fore solinne / varo satte sua sum tuu predikaro stola / annar høghra vaghin i þe palacio / oc annar a vinstra væghin /
'One person appeared awake, and not sleeping, as if she was in a palace. And in the furthest wall a large sun could be seen. Before the sun were set two pulpits, one against the right wall of the palace, the other against the left.' (*Birgitta-autograferna*)

Whereas the baseline is obviously useless in this case, the model has picked up enough to correctly identify a couple of L_0s in the material – a recall of just over 5% in this case corresponds to only 3 sentences. Of course, a model trained on the other documents, which *do* provide capitalization information, is likely to give too little weight to cases where the alternative strategies are used alone. This suggests that a model that cannot use capitalization as a feature should do better on a text like fold 1. Indeed, the 'no capitalization' model that fares poorly overall (Table 4), correctly identifies 10 sentence boundaries, a recall of almost 20%, without any loss in precision on this fold.

Although the text in fold 1 is too small to draw any hard conclusions, it does suggest a strategy for coping with the between document variation: using statistics over the text like the token-delimiter ratio for different markers, we can try to select a feature set that is likely to fit the data well. Working out the details of such a strategy and evaluating it will have to remain the subject of future work.

4 Conclusion and Outlook

The segmentation of historical texts into sentence-like units is a useful but hard task. In this paper, we have given a brief overview of the kind of boundary marking strategies we observe in our corpus of Old Swedish. Furthermore, we showed that a model that combines clues from punctuation, capitalization and lexical content is able to improve upon a simple capitalization baseline, especially in terms of precision.

As it stands, the segmentation quality of the models described in this paper is lacking. We can distinguish three use scenarios: for the presentation of corpus query results to users, as a first step in an automatic processing pipeline, and as a preprocessing step in a manual annotation task. In the latter scenario, the segmentation can be adjusted by the annotator and we judge the segmentation quality to be high enough to be helpful. In the first two cases, however, quality is crucial and currently too low.

The main finding of the paper is that the variation between documents is a real obstacle towards better performance. This variation comes in two flavours. First, we have variation in boundary marking strategies. In future work, we hope to be able to use statistics over the occurrence of known boundary markers in a text to choose a model that is likely to give good results for that particular style of marking. Secondly, there is variation in the spelling of words, which means that the models have a hard time picking up lexical clues. It is remarkable, however, that even the simple character mappings used in our experiments make a noticeable difference. In future work, we will investigate more careful and principled spelling normalization to address this problem better.

In the presentation use case mentioned above, we could sacrifice some recall for precision. At a recall of .5, segments will be on average twice as long as they should be, but this is still better than presenting an unsegmented text to the user. Trading recall for precision is generally possible in statistical models, and we will look at methods for boosting precision in future work.

Acknowledgments

We thank three anonymous reviewers for their comments. The research presented here is carried out in the context of the Centre for Language Technology of the University of Gothenburg and Chalmers University of Technology:–

References

Adesam, Y., Ahlberg, M., and Bouma, G. (2012). *bokstaffua, bokstaffwa, bokstafwa, bokstaua, bokstawa...* Towards lexical link-up for a corpus of Old Swedish. In Jancsary, editor, *Empirical Methods in Natural Language Processing: Proceedings of KONVENS 2012 (LThist 2012 workshop)*, page 365–369, Vienna.

Evert, S. (2005). *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, IMS Stuttgart.

Gillick, D. (2009). Sentence boundary detection and the problem with the U.S. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 241–244, Boulder, Colorado. Association for Computational Linguistics.

Gotoh, Y. and Renals, S. (2000). Sentence boundary detection in broadcast speech transcripts. In *ASR2000 - Automatic Speech Recognition: Challenges for the new Millenium*, pages 228–235, Paris, France.

Haug, D. T. T., Jøhndal, M., Eckhoff, H. M., Welo, E., Hertzenberg, M. J. B., and Müth, A. (2009). Computational and linguistic issues in designing a syntactically annotated parallel corpus of indo-european languages. *Traitement Automatique des Langues*, 50.

Höder, S. (2011). *Phrases and Clauses Tagging Manual for syntactic analyses of Old Nordic texts encoded as Menotic XML documents (PaCMan)*. University of Hamburg, Hamburg. Version 2.0.

Huang, H.-H., Sun, C.-T., and Chen, H.-H. (2010). Classical Chinese sentence segmentation. In *CIPS-SIGHAN Joint Conference on Chinese Language Processing*, pages 15–23.

Kiss, T. and Strunk, J. (2006). Unsupervised multilingual sentence boundary detection. *Computational Linguistics*, 32(4):485–525.

Liu, Y. and Shriberg, E. (2007). Comparing evaluation metrics for sentence boundary detection. In *ICASSP*.

Liu, Y., Stolcke, A., Shriberg, E., and Harper, M. (2005). Using Conditional Random Fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 451–458, Ann Arbor, Michigan. Association for Computational Linguistics.

Loman, B. and Jørgensen, N. (1971). *Manual for analys och beskrivning av makrosyntagmer*. Studentlitteratur, Lund.

Mikheev, A. (2002). Periods, capitalized words, etc. *Computational Linguistics*, 28(3):289–318.

Petran, F. (2012). Studies for segmentation of historical texts: Sentences or chunks? In Mambrini, F., Passarotti, M., and Sporleder, C., editors, *Proceedings of the Second Workshop on Annotation of Corpora for Research in the Humanities*, pages 75–86, Lisbon.

Read, J., Dridan, R., Oepen, S., and Solberg, L. J. (2012). Sentence boundary detection: A long solved problem? In *Proceedings of COLING 2012: Posters*, pages 985–994, Mumbai, India. The COLING 2012 Organizing Committee.

Stevenson, M. and Gaizauskas, R. (2000). Experiments on sentence boundary detection. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 84–89, Seattle, Washington, USA. Association for Computational Linguistics.

Svensson, L. (1974). *Nordisk Paleografi*. Number 28 in Lunda studier i nordisk språkvetenskap, serie A. Studentlitteratur, Lund.