SWEGRAM – A Web-Based Tool for Automatic Annotation and Analysis of Swedish Texts

Jesper Näsman
Linguistics and Philology
Uppsala University
jesper.nasman@lingfil.uu.se

Beáta Megyesi
Linguistics and Philology
Uppsala University
beata.megyesi@lingfil.uu.se

Anne Palmér
Scandinavian Languages
Uppsala University
anne.palmer@nordiska.uu.se

Abstract

We present SWEGRAM, a web-based tool for the automatic linguistic annotation and quantitative analysis of Swedish text, enabling researchers in the humanities and social sciences to annotate their own text and produce statistics on linguistic and other text-related features on the basis of this annotation. The tool allows users to upload one or several documents, which are automatically fed into a pipeline of tools for tokenization and sentence segmentation, spell checking, part-of-speech tagging and morpho-syntactic analysis as well as dependency parsing for syntactic annotation of sentences. The analyzer provides statistics on the number of tokens, words and sentences, the number of parts of speech (PoS), readability measures, the average length of various units, and frequency lists of tokens, lemmas, PoS, and spelling errors. SWEGRAM allows users to create their own corpus or compare texts on various linguistic levels.

1 Introduction

Although researchers in natural language processing have focused for decades on the development of tools for the automatic linguistic analysis of languages and state-of-the-art systems for linguistic analysis have achieved a high degree of accuracy today, these tools are still not widely used by scholars in the humanities and social sciences. The main reason is that many of the tools require programming skills to prepare and process texts. Furthermore, these tools are not linked in a straightforward way to allow the annotation and analysis on different linguistic levels that could be used easily in data-driven text research.

In this paper, we present SWEGRAM, a web-based tool for the automatic linguistic annotation and quantitative analysis of Swedish text, which allows researchers in the humanities and social sciences to annotate their own text or create their own corpus and produce statistics on linguistic and other text-related features based on the annotation. SWEGRAM requires no previous knowledge of text processing or any computer skills, and is available online for anyone to use.[1]

We start with a brief overview of some important infrastructural tools for processing language data. In Section 3 we give an introduction to SWEGRAM along with our goals and considerations in developing the web-based tool. Following this introductory section, we present the components used for the linguistic annotation on several levels, and the format of the data representation. We then give an overview of quantitative linguistic analysis, providing statistics on various linguistic features for text analysis. In Section 4 we describe a linguistic study of student essays to illustrate how SWEGRAM can be used by scholars in the humanities. Finally, in Section 5, we conclude the paper and identify some future challenges.

2 Background

To make language technology applications available and useful to scholars of all disciplines, in particular researchers in the humanities and social sciences has attracted great attention in the language technology community in the past years. One aim is to create language resources and tools that are readily available for automatic linguistic analysis and can help in quantitative text analysis. Important resources are corpora and lexicons of various kinds. Basic tools usually include a tokenizer for the automatic segmentation of tokens and sentences, a lemmatizer for finding the base form of words, a part-of-speech (PoS) tagger to annotate the words with their PoS and morpholog-

[1] http://stp.lingfil.uu.se/swegram/

Proceedings of the 21st Nordic Conference of Computational Linguistics, pages 132–141,
Gothenburg, Sweden, 23-24 May 2017.

ical features, and a syntactic parser to annotate the syntactic structure of the sentence.

Creating infrastructure for language analyis is not new and several projects have been focusing on developing on-line services for collection, annotation and/or analysis of language data with joint effort from the LT community. One of the important projects is the European Research Infrastructure for Language Resources and Technology CLARIN[2] with nodes in various countries, such as the Swedish SWE-CLARIN[3]. During the past years, we have seen a noticable increase in web-services allowing storage, annotation and/or analysis of data for various languages. Such example include LAP: The CLARINO Language Analysis Portal that was developed to allow large-scale processing service for many European languages (Kouylekov et al., 2014; Lapponi et al., 2014); WebLicht, a web-based tool for semi-annotation and visualization of language data (Hinrichs et al., 2010; CLARIN-D/SfS-Uni. Tübingen, 2012); The Australian project Alveo: Above and Beyond Speech, Language and Music infrastructure, a virtual lab for human communication science, for easy access to language resources that can be shared with workflow tools for data processing (Estival and Cassidy, 2016).

Many language technology tools are readily available as off-the-shelf packages and achieve a high degree of accuracy, including the analysis of Swedish text. A pipeline in which standard annotation tools can be run on-line was recently established through SPARV (Borin et al., 2016) at the Swedish Language Bank (Språkbanken),[4] for the linguistic analysis of uploaded text, including tokenization, lemmatization, word senses, compound analysis, named-entity recognition, PoS and syntactic analysis using dependency structures. Users can access the annotation directly online, or download the results as an XML document. The goal is to provide linguistic annotation and allow further analysis using Språkbanken's own corpus search tool, Korp (Borin et al., 2012).[5]

Many tools are available for various types of text analysis. These include search programs for analyzing specific resources or corpora. Examples include Xaira,[6] an open source software package that supports indexing and analysis of corpora using XML, which was originally developed for the British National Corpus; the BNCWeb (Hoffmann et al., 2008), a web-based interface for the British National Corpus; or Korp (Borin et al., 2012), for searches of Swedish corpora. Other popular tools are concordance programs, such as AntConc,[7] Webcorp[8] and ProtAnt (Anthony and Baker, 2015), which also displays other text related features such as frequencies, collocations and keywords. WordSmith Tools (Scott, 2016) is also commonly used for text analysis, allowing the creation of word lists with frequencies, concordance lists, clusters, collocations and keywords.

Next, we describe SWEGRAM, a publicly available on-line tool for corpus creation, annotation and data-driven analysis of Swedish text.

3 SWEGRAM

The main goal of SWEGRAM is to provide a simple web-based tool that allows linguistic annotation and quantitative analysis of Swedish text without any expert knowledge in natural language processing. SWEGRAM consists of two separate web-based applications: annotation and analysis. In the web-based interface, users can upload one or several text files of their choice and receive the annotated text(s), which can be sent for further text analysis, as specified by the user.

The annotation includes tokenization and sentence segmentation, normalization in terms of spelling correction, PoS tagging including morphological features, and dependency parsing to represent the syntactic structure of the sentence. The annotation tool can be used to annotate individual texts or create a large collection of annotated texts, a corpus.

Once the data set is uploaded and annotated, the analyzer provides information about the number of tokens, words, and sentences; the distribution of PoS and morphological features; various readability measures; average length of different units (such as words, tokens, sentences); frequency lists and spelling errors.

In developing SWEGRAM, we wanted to create a tool with open source components that was freely accessible, where users can upload any text without it being saved by the system. Another important goal was to build a modular system

[2]https://www.clarin.eu/

[3]https://sweclarin.se/eng/about

[4]https://spraakbanken.gu.se/sparv/

[5]https://spraakbanken.gu.se/korp/

[6]http://projects.oucs.ox.ac.uk/xaira/Doc/refman.xml

[7]http://www.laurenceanthony.net/software.html

[8]http://www.webcorp.org.uk/live/

in which the components involved can be easily changed as better models are developed, while individual components can be built on one another with a simple representation format that is easy to understand.

The pipeline handling linguistic annotation is written mainly in Python, and the user interface was developed using regular HTML, CSS and JavaScript. The backend of the web interface was developed using the Django web framework. Next, we will describe the components included for annotation and analysis in more detail.

3.1 Automatic Annotation

In order to automatically process and annotate texts, we use state-of-the-art natural language processing tools trained on Swedish standard texts with a documented high degree of performance. The annotation pipeline is illustrated in Figure 1. When a file is uploaded, the document is preprocessed by converting the file into a plain text format. The text is segmented into sentences and tokens by a tokenizer and misspelled tokens are corrected for spelling errors by a normalizer. The corrected text is run through a PoS tagger and lemmatizer to get the base form of the words and their correct PoS and morphological annotation given the context. Finally, the sentences are syntactically analyzed by a parsing module using dependency analysis. The following subsections contain descriptions of each of these modules.

3.1.1 Preprocessing

In many cases, SWEGRAM does not require any preprocessing of documents. Users can upload documents in formats such as DOC, DOCX and RTF and the document is automatically converted into a plain text file encoded in UTF-8, which is what the annotation pipeline requires as input. The text is converted using unoconv,[9] which can handle any format that LibreOffice is able to import.

3.1.2 Tokenization

Tokenization is used to separate the words from punctuation marks and segment the sentences. Two tokenizers were considered for SWEGRAM: the tokenizer written in Java and used in the PoS tagger Stagger (Östling, 2013) and the Svannotate tokenizer, originally developed for the Swedish Treebank (Nivre et al., 2008). A comparison was made between these tokenizers, and only a few differences were found, since both tokenizers achieved similar results. However, while Svannotate is an independent, rule-based tokenizer written in Python, Stagger's tokenizer is built into the PoS tagger. We chose to include Svannotate for modularity and consistency in the pipeline since it is written in Python, like the rest of SWEGRAM.

In evaluating Svannotate to tokenize student writings (Megyesi et al., 2016), errors that occurred were due in part to the inconsistent use of punctuation marks – for example, when a sentence does not always end with an appropriate punctuation mark, either because abbreviations are not always spelled correctly or a new sentence does not always begin with a capital letter.

Since the annotation pipeline is modular, users have the option of tokenizing a text, manually correcting it and then using the corrected version for the remaining steps.

3.1.3 Normalization

After tokenization and sentence segmentation, normalization is carried out in the form of spelling correction, including correction of erroneously split compounds. Since there is no open source, state-of-the-art normalizer that is readily available for Swedish, we used a modified version of HistNorm (Pettersson et al., 2013) for spelling correction. HistNorm was originally developed to transform words in historical texts that had substantial variation in possible spellings of their modern variant using either Levenshtein-based normalization or normalization based on statistical machine translation (SMT). When used on historical data, HistNorm achieves accuracy of 92.9% on Swedish text, based on SMT. For texts written by students, however, we found that the Levenshtein-based normalization gave better results.

One type of spelling error that occurs frequently in Swedish is erroneously split compounds, that is, compounds that are split into two or more words instead of written as one word. If we consider the Swedish compound *kycklinglever* (chicken liver), erroneously splitting the words would form the two words *kyckling* (chicken) and *lever* (is alive). This significantly alters the meaning of the phrase and will affect the final output of the annotation, making the statistical analysis less accurate. Addressing these errors can lead to an improved annotation performance. This problem is addressed using a rule-based system as described by (Öhrman, 1998). Because of the PoS tags

[9] https://github.com/dagwieers/unoconv

Figure 1: Screenshot of the web-based annotation interface.

rules for identifying split compounds for each token, PoS tagging has to be performed prior to correcting compounds. The text is then tagged again using the corrected compounds. We will return to how these types of corrections are represented while still keeping the original tokens in Section 3.1.6.

Further analysis and improvement are needed to adapt this normalization tool to texts written in less standard Swedish for a higher degree of accuracy.

3.1.4 Morpho-Syntactic Annotation

For the PoS and morphological annotation of the normalized texts, we use two types of annotation. One is based on the universal PoS tagset,[10] which consists of 17 main PoS categories: adjective, adposition, adverb, auxiliary, coordinating conjunction, determiner, interjection, noun, numeral, particle, pronoun, proper noun, punctuation, subordinating conjunction, symbol, verb and others with their morphological features. The other tagset used is the Stockholm-Umeå Corpus tagset (Gustafson-Capková and Hartmann, 2006), which contains 23 main PoS categories.

We compared two commonly used PoS taggers for Swedish, HunPos (Halácsy et al., 2007) and Stagger (Östling, 2013), and evaluated their performance on our test data. Both taggers used models trained on what is normally used as a standard corpus for Swedish, the Stockholm Umeå Corpus (Gustafson-Capková and Hartmann, 2006). The accuracy of these taggers when trained and evaluated on SUC 2.0 is very similar, 95.9% for HunPos (Megyesi, 2008) and 96.6% for Stagger (Östling, 2013). Testing these taggers on the Uppsala Corpus of Student Writings (Megyesi et al., 2016) using SUC models, Stagger performed slightly better. Another advantage of Stagger is that it can also perform lemmatization.

However, we ultimately decided to use a reimplementation of Stagger, the tagger called Efficient Sequence Labeler (efselab),[11] as the default tagger. This, like Stagger, uses an averaged perceptron learning algorithm, but Efselab has the advantage that it performs PoS tagging significantly faster (about one million tokens a second) while achieving similar performance results as Stagger.

3.1.5 Syntactic Annotation

The final step in the annotation pipeline is the syntactic annotation in terms of dependency structure. We apply universal dependencies (UD) (Nivre et al., 2016) to mark syntactic structures and relations where one word is the head of the sentence, attached to a ROOT, and all other words are dependent on another word in a sentence. Dependency relations are marked between content words while function words are direct dependents of the most closely related content word. Punctuation marks are attached to the head of the clause or phrase to which they belong. The UD taxonomy distinguishes between core arguments such as subjects, direct and indirect objects, clausal complements, and other non-core or nominal dependents. For a detailed description of the dependency structure and annotation, we refer readers to the UD website.[12]

To annotate the sentences with UD, we use MaltParser 1.8.1 (Nivre et al., 2006), along with a model trained on the Swedish data with Universal Dependencies (UD). Since parser input needs to be in the form of the universal tagset, the tags need to be converted. This conversion is carried out using a script that comes with efselab, which converts SUC to UD.

Since UD was developed in our field of natural language processing only recently, it has not been used widely by scholars outside our community. In the near future, we will experiment with various types of syntactic representation.

3.1.6 Format

In order to make it easy for scholars in the humanities to interpret the annotated texts, we chose the CoNLL-U tab-separated format[13] instead of an XML-based representation. Sentences consist of one or more lines of words where each line represents a single word/token with a series of 11 fields

[10] http://universaldependencies.org/u/pos/
[11] https://github.com/robertostling/efselab
[12] http://universaldependencies.org/
[13] http://universaldependencies.org/format.html

with separate tabs for various annotation types. Table 1 describes the fields that represent the analysis of each token. New sentences are preceded by a blank line, which marks sentence boundaries. Comment lines starting with hash (#) are also allowed and may be used for metadata information (such as sentence numbering) for the sentence following immediately. All annotations are encoded in plain text files in UTF-8.

In Table 2 an example is provided of an annotated text in the CoNLL-U format. In this example, the original text contains a spelling mistake, *vekan*, corrected as *veckan* in the column NORM, where the corrected form is analyzed. The example sentence also contains an erroneously split compound – *Syd Korea* which should be written as one word, *Sydkorea*. The corrected word is given the index numbers of the two original words, in this case 4-5, where the corrected version is analyzed linguistically while the original forms are left as they are without any further analysis.

Text containing metadata has been an important factor in the development of SWEGRAM. Metadata containing information about the text such as the author's age, gender, geographic area or type of text can be parsed and used during analysis, allowing users to filter their texts based on the metadata provided, and produce statistics on the features of the particular text(s). The metadata should be represented in the format <feature1, feature2 ... featureN>. Development is currently under way to allow metadata of any type (defined by the user) to be used in annotation and analysis.

3.1.7 Web-based Annotation Tool

The web-based annotation tool is illustrated in Figure 2. Users can upload one or several texts and annotate them.

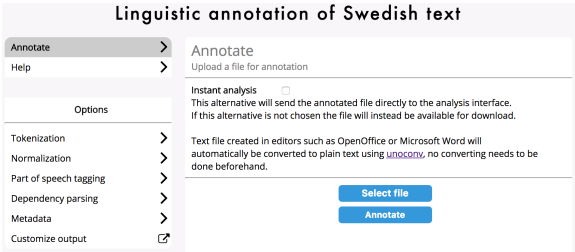

Figure 2: Screenshot of the web-based annotation interface.

Modularity has been an important factor in developing the annotation tool. Any module can be deactivated, which enables users to exclude some part of the annotation if they wish and use their own annotation instead. For example, users can upload a text that is already tokenized in order to annotate it with PoS and syntactic features. After tokenization, normalization can also be carried out in the form of spell checking and correction of erroneously split compound words, or a text that is already corrected can be uploaded. Similarly, users could correct the PoS annotation given by the tool and run the syntactic analyzer on the corrected PoS tagged data. Users are thus free to decide which particular tools are needed, and the subsequent linguistic annotation is based on corrected, normalized forms, which could help improve the performance of subsequent steps since corrected data are used.

Each module may include several algorithms and models depending on the corpus data the models were trained on. We include the most frequently used models with the highest accuracy on standard Swedish, which were evaluated and published previously.

Moreover, the pipeline is built in such a way that new, better analyzers can be plugged into the system. It is also possible to select different models for the PoS tagger and the syntactic parser, but currently only one model is provided for each, both based on Stockholm-Umeå Corpus (SUC) 3.0 (Gustafson-Capková and Hartmann, 2006) and previously evaluated with a documented high degree of accuracy. However, one restriction in choosing syntactic annotation (the parser and parser model) is that only the PoS model that the parser was trained on may be run during the PoS tagging module to get consistent annotation.

Another important factor was that the format should be readable and easy to understand so that users can manually examine the data annotated. The results are made available to users in the form of a downloadable plain text file encoded in UTF-8 or shown directly on the web-page. In contrast to formats like SGML or XML, the CoNLL-U format, which is tab-separated with one token per line and has various linguistic fields represented in various columns, is well suited for our purposes. The format with fields separated by tabs allows users to import their file in Excel or another tool of their choice to carry out further quantitative analysis.

Since the corpus format allows several types of

FEATURE	Description
TEXT ID	Paragraph-sentence index, integer starting at 1 for each new paragraph and sentence
TOKEN ID	Token index, integer starting at 1 for each new sentence; may be a range for tokens with multiple words
FORM	Word form or punctuation symbol
NORM	Corrected/normalized token (e.g. in case of spelling error)
LEMMA	Lemma or stem of word form
UPOS	Part-of-speech tag based on universal part-of-speech tag
XPOS	Part-of-speech tag based on the Stockholm-Umeå Corpus; underscore if not available
XFEATS	List of morphological features for XPOS; underscore if not available
UFEATS	List of morphological features for UPOS; underscore if not available
HEAD	Head of the current token, which is either a value of ID or zero (0)
DEPREL	Dependency relation to the HEAD (root iff HEAD = 0) based on the Swedish Treebank annotation
DEPS	List of secondary dependencies (head-deprel pairs)
MISC	Any other annotation

Table 1: Annotation representation format for each token and field.

TEXT ID	ID	FORM	NORM	LEMMA	UPOS	XPOS	XFEATS	UFEATS	HEAD	DEPREL	DEPS	MISC
2.4	1	Jag	Jag	jag	PRON	PN	UTR |SIN |DEF |SUB	Case=Nom|Definite=Def|Gender=Com|Number=Sing	0	root	_	I
2.4	2	var	var	vara	VERB	VB	PRT |AKT	Mood=Ind|Tense=Past|VerbForm=Fin|Voice=Act	1	acl	_	was
2.4	3	i	i	i	ADP	PP	_	_	4-5	case	_	in
2.4	4-5		Sydkorea	Sydkorea	PROPN	PM	NOM	Case=Nom	2	nmod	_	South Korea
2.4	4	Syd	Syd									South
2.4	5	Korea	Korea									Korea
2.4	6	förra	förra	förra	ADJ	JJ	POS |UTR/NEU |SIN |DEF |NOM	Case=Nom|Definite=Def|Degree=Pos|Number=Sing	7	det	_	last
2.4	7	vekan	veckan	vecka	NOUN	NN	UTR |SIN |DEF |NOM	Case=Nom|Definite=Def|Gender=Com|Number=Sing	4-5	nmod	_	week
2.4	8	.	.	.	PUNCT	MAD	_	_	1	punct	_	.

Table 2: Example of the extended CoNLL-U shared task format for the sentence *Jag var i Syd Korea förra vekan* (I was in South Korea last week). It contains one misspelled word, *veckan*, and one erroneously split compound, *Syd Korea* – South Korea, which should be a single compound word in Swedish. Note that the *MISC* column here is used to provide English translations for this table.

annotation by including additional columns, users can easily choose between them based on their desires or choose to have all annotations available.

3.2 Automatic Quantitative Analysis

Users can upload one or several annotated texts for further quantitative analysis. Statistics are calculated and shown on several levels: for all texts, and if the text file is divided into several subtexts, for each of these. Figure 3 illustrates the start page of the quantitative analysis where information is given about the number of uploaded texts, words, tokens and sentences.

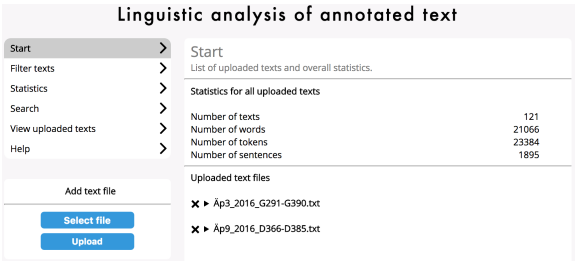

Figure 3: Automatic quantitative analysis.

The following features can be extracted automatically: number of tokens, words, sentences, texts and PoS; readability measures; average length of words, tokens, sentences, paragraphs and texts; frequency lists of tokens, lemmas and PoS; and spelling errors.

The statistical calculations are divided into three sections: general statistics, frequencies and spelling errors. General statistics provide users with the option of including statistics for all PoS or for specific ones, readability metrics in terms of LIX, OVIX and the nominal ratio, and frequencies of word length above, below or at a specific threshold value.

The frequencies section can provide users with frequency lists for all texts and for individual texts. These can be based on lemmas or tokens, with or without delimiters. In addition, the frequency lists can be sorted based on frequencies or words (lemmas or tokens) in alphabetical order. The frequency lists can also be limited to specific parts of speech.

The spelling errors section provides a list of spelling errors sorted by frequency, for all uploaded texts and for individual texts.

In addition, users can generate statistics by filtering the texts using metadata information. In order to do so, the uploaded texts have to be marked up with metadata as described in Section 3.1.6.

Given each field, the texts can be filtered based on the properties of the metadata. Examples of analyses are provided in the next section.

Users can also specify whether the output should be delivered as a downloadable file separated by tabs, which can be imported into other programs such as Excel for further analysis, or shown directly in the browser.

Separately from the statistics, users can also view the uploaded texts in their entirety and perform different types of searches in the annotated text. This includes searching for words, lemmas and PoS tags that either start with, end with, contain or exactly match a user-defined search query. The results are then printed and sorted according to what texts they appear in.

4 User Study

In this section we will demonstrate some of the possibilities of using SWEGRAM to analyze student writing as part of the national test carried out by school children in Sweden. We concentrate on two texts which are interesting to compare because they have some features in common but also differ in terms of the age of the writers, with the difference being three school years. Without making use of SWEGRAMs capacity to analyze extensive data, we simply want to demonstrate some features included in the tool and what they can show about the characteristics of the two texts.

Essay D245, from the last year of compulsory school, and essay C381, from the final year of upper secondary school, both represent the expository genre. Both essays have also been used as examples, benchmarks, of essays receiving the highest grade in the guide for assessing national tests. Therefore these two essays are both considered to be good writing for their respective school year. However, there is a three-year difference in age between the students, and the writing assignments given in the tests are different. Text D245 discusses a general subject, love. The introduction of the essay, translated into English, is: *Would you risk sacrificing your life for love, would you risk turning your entire existence upside down? The question is not easy to answer.* Text C381 is an expository essay on the fairytale *Sleeping Beauty*, which makes the subject more specific. The introduction to this essay is translated into English as: *Folk tales – anonymous stories that have been passed down from one generation to the next no matter where humans have lived /.../ Why is this old fairytale still widely read in society today?*.

We compare the documents in terms of different features that can give information relevant to text quality and writing development, such as lexical variation and specification, word frequencies, nominal ratio and distribution of parts-of speech.

Looking at the lexical variation, the two texts are about the same length; D245 has 790 words and C381 has 713 words. But the average word length of the text from upper secondary school is higher than that of the text from compulsory school, as shown in Table 3.

	D245	**C381**
Word length	4.70	5.66
Ovix	52.87	81.70
Nominal ratio	0.55	1.35
Sentence length	20.27	25.73

Table 3: Some measures from SWEGRAM.

These results indicate that C381 may be more specified and lexically varied than D245, since longer words correlate with specification and variation in a text (Hultman and Westman, 1977; Vagle, 2005). Lexical variation in a text can also be measured by Ovix, a word variation index (Hultman and Westman, 1977). This measure shows the same tendencies: more variation in the text from upper secondary school.

The lexicon can further be studied using SWEGRAMs word frequency lists. In the list of nouns we look for long, compound nouns, since this is considered one feature of Swedish academic language. We find a number of these long words, several with more than 12 letters, in C 381. In D 245 there are a few compound nouns but none as long as this, which makes the lexicon of this text less specified and dense.

Nominal ratio is used to measure the nominality of the text. A high nominal ratio indicates high information load, whereas a low nominal ratio indicates a less dense text (Magnusson and Johansson Kokkinakis, 2011). Texts with high nominality are often conceived as having more of a written style, whereas lower values tend to give the text a more colloquial character. The difference in the nominal ratio for the two texts is substantial, 0.55 in D245 and as high as 1.35 in C381, as shown in Table 3. As a result, the essay from upper secondary school is considerably more nominal, has a

higher information load and presumably has more of a written style than the essay from compulsory school. The surprisingly high value of the nominal ratio in C381 could partly be explained by the fact that there are several references to other works in this text, and these include long nominal phrases.

D245	C381
VB (17.38%)	NN (19.82%)
NN (12.33%)	VB (13.34%)
PN (11.66%)	PP (11.27%)
AB (10.87%)	AB (7.12%)
PP (8.30%)	JJ (6.99%)

Table 4: The five most frequently occurring parts of speech.

A look at the parts of speech used most frequently shows that D245 is rich in verbs and pronouns, parts of speech that characterize a colloquial style; see Table 4. C381, on the other hand, has high proportions of nouns and prepositions, which are important words in forming nominal phrases.

Table 3 shows that there is a difference in the average sentence length in the two essays: 20.27 words in D245 and 25.73 in C381. Since longer sentences may contain more clauses than shorter ones, this result indicates that the syntax of the essay from upper secondary school may be more complex that in D 245. The hypothesis can be controlled by a frequency list of conjunctions and subjunctions, words that connect clauses. In D245 there are six different conjunctions and three different subjunctions, a total of nine connectives of this kind. In C381 there are eight different conjunctions and four subjunctions, a total of twelve different words. So the variation in connectives is more important in C381. The distribution of parts of speech also shows that conjunctions and subjunctions occur more frequently in C381 (KN + SN 7.12 %) than in D245 (KN + SN 5.54 %), which supports the hypothesis.

In summary, the analysis shows considerable differences between the two essays, as regards the lexicon, distribution of PoS and syntax. However, the result should not be interpreted in relation to the writing competence or writing development shown in the student texts. The purpose is to show the potential of analyses made with SWEGRAM without using the appropriate amount of data.

5 Conclusion

We presented a web-based interface for the automatic annotation and quantitative analysis of Swedish. The web-based tool enables users to upload a file, which is then automatically fed into a pipeline of tools for tokenization and sentence segmentation, spell checking, PoS tagging and morpho-syntactic analysis as well as dependency parsing for syntactic annotation of sentences. Users can then send the annotated file for further quantitative analysis of the linguistically annotated data. The analyzer provides statistics about the number of tokens, words, sentences, number of PoS, readability measures, average length of various units (such as words, tokens and sentences), frequency lists of tokens, lemmas and PoS, and spelling errors. Statistics can be also extracted based on metadata, given that metadata are defined by the user.

The tool can be easily used for the analysis of a single text, for the comparison of several texts, or for the creation of an entire corpus of the user's choice by uploading a number of text documents. The tool has been used succesfully in the creation of the Uppsala Corpus of Student Writings (Megyesi et al., 2016). Since SWEGRAM will be used to create corpora, the possibility of customizing the content and format of metadata is something that could be beneficial to users and will be implemented in the near future.

The tools are readily available and can be used by anyone who is interested in the linguistic annotation of Swedish text. As better models for standard Swedish are presented, our intention is to include them in the interface along with the old models to allow comparative studies. Our priority for further improvement is the normalization tool since there is no readily available open source tool for automatic spelling and grammar correction of Swedish. In addition, we would like to implement a visualization tool of the linguistic analysis, especially syntax, which will also facilitate syntactic searches.

Acknowledgments

This project was supported by SWE-CLARIN, a Swedish consortium in Common Language Resources and Technology Infrastructure (CLARIN) financed by the Swedish Research Council for the period 2014–2018.

References

Laurence Anthony and Paul Baker. 2015. ProtAnt: A tool for analysing the prototypicality of texts. *International Journal of Corpus Linguistics*, 20(3):273–292.

Lars Borin, Markus Forsberg, and Johan Roxendal. 2012. Korp – the corpus infrastructure of Språkbanken. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, LREC 2012, page 474478.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Anne Schumacher, and Roland Schäfer. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *SLTC 2016*.

CLARIN-D/SfS-Uni. Tübingen. 2012. WebLicht: Web-Based Linguistic Chaining Tool. Online. Date Accessed: 28 Mar 2017. URL https://weblicht.sfs.uni-tuebingen.de/ .

Dominique Estival and Steve Cassidy. 2016. Alveo: Above and beyond speech, language and music, a virtual lab for human communication science. Online. Date Accessed: 28 Mar 2017. URL http://alveo.edu.au/about/.

Sofia Gustafson-Capková and Britt Hartmann, 2006. *Documentation of the Stockholm - Umeå Corpus*. Stockholm University: Department of Linguistics.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: An open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erhard W. Hinrichs, Marie Hinrichs, and Thomas Zastrow. 2010. Weblicht: Web-based LRT services for German. In *Proceedings of the ACL 2010 System Demonstrations*, pages 25–29.

Sebastian Hoffmann, Stefan Evert, Nicholas Smith, David Lee, and Ylva Berglund Prytz. 2008. *Corpus Linguistics with BNCweb – A Practical Guide*. Frankfurt am Main: Peter Lang.

Tor G. Hultman and Margareta Westman. 1977. *Gymnasistsvenska*. LiberLäromedel, Lund.

Milen Kouylekov, Emanuele Lapponi, Stephan Oepen, Erik Velldal, and Nikolay Aleksandrov Vazov. 2014. LAP: The language analysis portal. Online. Date Accessed: 28 Mar 2017. URL http://www.mn.uio.no/ifi/english/research/projects/-clarino/.

Emanuele Lapponi, Erik Velldal, Stephan Oepen, and Rune Lain Knudsen. 2014. Off-road laf: Encoding and processing annotations in nlp workflows. In *9th edition of the Language Resources and Evaluation Conference (LREC)*.

Ulrika Magnusson and Sofie Johansson Kokkinakis. 2011. Computer-Based Quantitative Methods Applied to First and Second Language Student Writing. In Inger Källström and Inger Lindberg, editors, *Young Urban Swedish. Variation and change in multilingual settings*, pages 105–124. Göteborgsstudier i nordisk språkvetenskap 14. University of Gothenburg.

Beáta Megyesi, Jesper Näsman, and Anne Palmér. 2016. The Uppsala corpus of student writings: Corpus creation, annotation, and analysis. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 3192–3199, Paris, France. European Language Resources Association (ELRA).

Beáta Megyesi. 2008. *The Open Source Tagger HunPoS for Swedish*. Uppsala University: Department of Linguistics and Philology.

Joakim Nivre, Johan Hall, and Jens Nilsson. 2006. Maltparser. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, LREC '06, pages 2216–2219.

Joakim Nivre, Beáta Megyesi, Sofia Gustafson-Capková, Filip Salomonsson, and Bengt Dahlqvist. 2008. Cultivating a Swedish treebank. In Joakim Nivre, Mats Dahllöf, and Beáta Megyesi, editors, *Resourceful Language Technology: A Festschrift in Honor of Anna Sågvall Hein*, pages 111–120.

Joakim Nivre, Željko Agić, Lars Ahrenberg, Maria Jesus Aranzabe, Masayuki Asahara, Aitziber Atutxa, Miguel Ballesteros, John Bauer, Kepa Bengoetxea, Yevgeni Berzak, Riyaz Ahmad Bhat, Eckhard Bick, Carl Börstell, Cristina Bosco, Gosse Bouma, Sam Bowman, Gülşen Cebirolu Eryiit, Giuseppe G. A. Celano, Fabricio Chalub, Çar Çöltekin, Miriam Connor, Elizabeth Davidson, Marie-Catherine de Marneffe, Arantza Diaz de Ilarraza, Kaja Dobrovoljc, Timothy Dozat, Kira Droganova, Puneet Dwivedi, Marhaba Eli, Tomaž Erjavec, Richárd Farkas, Jennifer Foster, Claudia Freitas, Katarína Gajdošová, Daniel Galbraith, Marcos Garcia, Moa Gärdenfors, Sebastian Garza, Filip Ginter, Iakes Goenaga, Koldo Gojenola, Memduh Gökrmak, Yoav Goldberg, Xavier Gómez Guinovart, Berta Gonzáles Saavedra, Matias Grioni, Normunds Grūzītis, Bruno Guillaume, Jan Hajič, Linh Hà M, Dag Haug, Barbora Hladká, Radu Ion, Elena Irimia, Anders Johannsen, Fredrik Jørgensen, Hüner Kaşkara, Hiroshi Kanayama, Jenna Kanerva, Boris Katz, Jessica Kenney, Natalia Kotsyba, Simon Krek, Veronika Laippala, Lucia Lam, Phng Lê Hng, Alessandro Lenci, Nikola Ljubešić, Olga Lyashevskaya, Teresa Lynn, Aibek Makazhanov, Christopher Manning, Cătălina Mărănduc, David Mareček, Héctor Martínez Alonso, André Martins, Jan Mašek, Yuji Matsumoto, Ryan McDonald, Anna

Missilä, Verginica Mititelu, Yusuke Miyao, Simonetta Montemagni, Keiko Sophie Mori, Shunsuke Mori, Bohdan Moskalevskyi, Kadri Muischnek, Nina Mustafina, Kaili Müürisep, Lng Nguyn Th, Huyn Nguyn Th Minh, Vitaly Nikolaev, Hanna Nurmi, Petya Osenova, Robert Östling, Lilja Øvrelid, Valeria Paiva, Elena Pascual, Marco Passarotti, Cenel-Augusto Perez, Slav Petrov, Jussi Piitulainen, Barbara Plank, Martin Popel, Lauma Pretkalnia, Prokopis Prokopidis, Tiina Puolakainen, Sampo Pyysalo, Alexandre Rademaker, Loganathan Ramasamy, Livy Real, Laura Rituma, Rudolf Rosa, Shadi Saleh, Baiba Saulīte, Sebastian Schuster, Wolfgang Seeker, Mojgan Seraji, Lena Shakurova, Mo Shen, Natalia Silveira, Maria Simi, Radu Simionescu, Katalin Simkó, Mária Šimková, Kiril Simov, Aaron Smith, Carolyn Spadine, Alane Suhr, Umut Sulubacak, Zsolt Szántó, Takaaki Tanaka, Reut Tsarfaty, Francis Tyers, Sumire Uematsu, Larraitz Uria, Gertjan van Noord, Viktor Varga, Veronika Vincze, Lars Wallin, Jing Xian Wang, Jonathan North Washington, Mats Wirén, Zdeněk Žabokrtský, Amir Zeldes, Daniel Zeman, and Hanzhi Zhu. 2016. Universal dependencies 1.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics, Charles University in Prague.

Lena Öhrman, 1998. *Felaktigt särskrivna sammansättningar*. Stockholm University, Department of Linguistics.

Robert Östling. 2013. Stagger: An open-source part of speech tagger for Swedish. *Northern European Journal of Language Technology*, 3:1–18.

Eva Pettersson, Beáta Megyesi, and Joakim Nivre. 2013. Normalisation of historical text using context-sensitive weighted Levenshtein distance and compound splitting. In *Proceedings of the 19th Nordic Conference of Computational Linguistics*, NODALIDA '13.

Mike Scott, 2016. *WordSmith Tools Version 7*. Stroud: Lexical Analysis Software.

Wenche Vagle. 2005. Tekstlengde + ordlengdesnitt = kvalitet? Hva kvantitative kriterier forteller om avgangselevenas skriveprestasjoner. In Kjell Lars Berge, Siegfred Evensen, Frydis Hertzberg, and Wenche. Vagle, editors, *Ungdommers skrivekompetanse, Bind 2. Norskexamen som tekst*. Universitetsforlaget.