GÖTEBORGS UNIVERSITET
INST FÖR SVENSKA SPRÅKET

GU-ISS-2019-03

Assessing the quality of Språkbanken's annotations

Peter Ljunglöf, Niklas Zechner, Luis Nieto Piña,

Yvonne Adesam, Lars Borin

Forskningsrapporter från institutionen för svenska språket, Göteborgs universitet
Research Reports from the Department of Swedish

ISSN 1401-5919

www.svenska.gu.se/publikationer/GU-ISS

Assessing the quality of Språkbanken's annotations

Peter Ljunglöf, Niklas Zechner, Luis Nieto Piña,
Yvonne Adesam, Lars Borin

2019-06-10

Abstract

Most of the corpora in Språkbanken Text consist of unannotated plain text, such as almost all newspaper texts, social media texts, novels and official documents. We also have some corpora that are manually annotated in different ways, such as Talbanken (annotated for part-of-speech and syntactic structure), and the Stockholm Umeå Corpus (annotated for part-of-speech).

Språkbanken's annotation pipeline *Sparv* aims to automatise the work of automatically annotating all our corpora, while still keeping the manual annotations intact. When all corpora are annotated, they can be made available, e.g., in the corpus searh tools *Korp* and *Strix*.

Until now there has not been any comprehensive overview of the annotation tools and models that *Sparv* has been using for the last eight years. Some of them have not been updated since the start, such as the part-of-speech tagger *Hunpos* and the dependency parser *MaltParser*. There are also annotation tools that we still have not included, such as a constituency-based parser.

Therefore Språkbanken initiated a project with the aim of conducting such an overview. This document is the outcome of that project, and it contains descriptions of the types of manual and automatic annotations that we currently have in Språkbanken, as well as an incomplete overview of the state-of-the-art with regards to annotation tools and models.

Contents

1 Introduction

1.1 Why evaluation?

The ultimate purpose of evaluation is in Språkbanken's case to improve our research infrastructure so that it better answers to the requirements put on it by its users. Given the need to operationalise the means for gauging the degree of attainment of this purpose, we too often end up with ritualised evaluation regimes based on "gold standards" which for largely opportunistic purposes are recycled far beyond their applicability and/or with any caveats concerning their accuracy discarded along the way. This points to the need of investigating the relationship between the (fairly low-level) linguistic phenomena normally targeted in NLP evaluation exercises and the requirements that our users will have on the data and tools that we provide in the infrastructure.

However, to do this properly is a much larger assignment that we have had time for in this project, so this document should be seen as a sketch, focusing on asking questions rather than answering them.

1.2 What is quality?

Evaluation is in some respect carried out to explore the annotation quality, and therefore we need to define what we mean by quality. There are at least three aspects of quality that are of importance for annotation, and they are *well-formedness*, *consistency*, and *soundness*.

Further reading

Adesam (2012) Regarding well-formedness, consistency, soundness, and quality with regards to treebanks (pages 31–32):

- "A treebank that is well-formed is complete in the sense that each token and each non-terminal node is part of a sentence-spanning tree, and has a label."
- "A consistent treebank is consistently annotated, which means, e.g., that the same token sequence (or part-of-speech sequence or constituent sequence) is annotated in the same way across the treebank, given the same context."
- "[Soundness means] that the parallel treebank conforms to a linguistic theory. At the very least, it should conform to sound linguistic principles."
- "Quality is an issue that has been somewhat ignored in the area of computational linguistics. Often a resource, e.g., the Penn treebank, is used as the truth, without regard to what is in the data, or to issues that could have been done better if not pursuing getting good scores when evaluating against this particular resource. Or, as put by Levin, "[t]he people who say they love data the most seem to be the

most afraid of looking at it." (Levin, 2011, p. 14) It is of utmost importance to look at the data, and not just the global accuracy scores."

1.3 For whom are we doing this?

Who will be using the quality assessments?

(1) *Ourselves*: The quality assessments will be useful when we want to improve our annotation and other tools.

(2) *NLP researchers*: To have a standard to evaluate against, to influence other data providers to start assessing their quality, etc.

(3) *DH researchers*: To start thinking about which conclusions can be drawn from non-perfect data. To be aware that there are false negatives, i.e. that a search might not reveal all relevant results. (False positives are usually not as much of a problem, because you at least see and can dispose of them.)

(4) *All researchers*: Assessing the fit of our tools and resources to the research questions of the users of our research infrastructure.

Point 3 suggests that it is important to think about precision and recall. For Korp searches we should perhaps try to increase the recall, e.g., by annotating several pos/msd/senses for one word.

1.4 Things we are not covering

Here are some interesting related issues we didn't have the time to look into:

Evaluation of lexical resources. In what way can we evaluate our lexical resources (e.g., Saldo, Karp)? One possibility is coverage: how many percent of the word forms in a particular corpus is covered by the lexicon? In how many percent of these does the lexicon contain the correct pos/msd/sense?

Are we evaluating the corpus or the tool? This is more of a philosophical question – what is the relation between the corpus and the tool?

Presenting the confidence scores. If we have confidence scores for our annotations, how do we present these scores to a user? E.g., a normal Korp user, who might be a DH researcher or a linguist – what kind of information do they need to understand and be able to make decisions about the quality of the Korp search results?

2 Use cases

When we consider an annotation and an evalutation, we should ask ourselves some questions:

- In what way will the following annotation be useful for the intended audience?
- What quality measure will be useful?
- And what is the relationship – if any – between our conventional accuracy measures for the annotations and the quality measures presented to the intended audience?

One way to answer these questions is to come up with use cases, where we try to envision how a certain user would use our tools, and what information (about the annotation) they would be interested in.

The rest of this section consists of a list of possible use cases. Note that this list is not at all complete, and mainly consists of stubs (if anything).

The linguist is usually very good at spotting annotation errors or inconsistencies, and the challenge here is to convey a realistic picture of the quality of our resources and what kind of research a given level of quality realistically can support.

- *The synchronic linguist*
- *The corpus linguist* is much concerned with *representativity*, in the form of balanced corpora, cf. Brown, SUC, BNC.
- *The typological linguist* requires comparable data on many languages (ideally at least in the hundreds). This is a use case where UD and UPoST are of obvious utility, as well as interlinked lexical resources such as the IDS/LWT lists and multilingual WordNet and FrameNet.
- *The historical linguist* is in need of comparable diachronic data, with as fine a granularity as possible (decades rather than centuries), and preferably with rich and commensurable linguistic annotations. Quantitative historical linguistics requires large amounts of text.
- *The philologist* (possibly not an immediate priority for us.)
- *The sociolinguist*
- *Second-language acquisition* Text and vocabulary complexity is important in SLA research.

The historian will be interested in the *content* of texts, not their linguistic form. This use case will be similar to information access and the annotation challenge is mainly one of unifying textual elements across time, so that similar content can be found regardless of the linguistic form used to express it. This implies that evaluation procedures may be the same as those used in information access.

The conceptual historian wants to be able to track word usages over time, which implies that they need the same kind of support as historical linguists investigating lexical change, but probably presented in a different way.

The literary scholar wants to track topics both in single literary works and in more extensive bodies of literature. They would like to be able to discover allusions and

other forms of text reuse across works and ideally across languages, as well as meta-discourse about literature (notably literature reviews) and connections between literature and the real world. They are also interested in tracking the appearance and interaction of fictional characters in literary works. Finally, they would welcome maximally accurate methods for authorship attribution.

The political scientist requires access to the content of documents, much for the same reasons as the historian. In addition, they are often interested in tenor (sentiment/emotivity) and argumentation, as well as in inferring social networks from text content.

The psychologist is interested (at least) in different aspects of vocabulary, such as frequency of use, concreteness, sentiment/emotivity, age of acquisition, etc.

3 Confidence scores and evaluation

Some tools give an output which is more than a simple answer. It may give an estimate of the actual accuracy, that is, the chance that this token was correctly processed. This would likely be helpful for evaluating how well the tool is performing on different types of data. They might also give some more general measure of reliability, which can be translated into an accuracy, or something more limited, such as a classification of "sure" vs. "unsure", which we could also use to estimate the accuracy over a whole text.

Some tools give a set or ordered list of possible answers. This can also be helpful, especially in combination with other tools, but may be difficult to use as an immediate estimate of accuracy. If a tool always gives its top three guesses, that does not tell us anything about accuracy. If there is a variable-size unordered set of guesses, this can certainly tell use something about how "sure" the tool was, but it can also be difficult to use as input to another tool.

3.1 Combining confidence scores

We may need to combine the results of several tools in both serial and parallel ways.

If we have several tools in a pipeline, we will almost inevitably have tools that depend on previous ones to give accurate results. Suppose tool A uses the raw data and tool B uses the result from A. Let $\text{acc}(x, y, z, \ldots)$, where $(x, y, z, \ldots)$ are tools, be the accuracy of a sequence of tools.

To a first approximation, we can assume that if A goes wrong, B will also go wrong. In this case, we can assume that $\text{acc}(A, B) = \text{acc}(A) \times \text{acc}(1, B)$, where $acc(1, B)$ represents the accuracy of B given that it has completely correct input data.

If B has a limited number of possible outputs, then it would likely have a greater than zero accuracy even when given completely incorrect input; we can assume that

$\text{acc}(0, B) > 0$. In that case,

$$\text{acc}(A, B) = \text{acc}(A) \times \text{acc}(1, B) + (1 - \text{acc}(A)) \times \text{acc}(0, B)$$

In some cases, B might be only partially based on A, and partially on the original data, or some earlier step in the process. This should be be possible to test, finding an estimate for $\text{acc}(1, 0, B)$, etc.

In some cases, we have multiple tools for the same task, so we are able to combine them in parallel. The general problem does not have a solution; the combination of any two nontrivial probabilities could in principle be any probability. This may have to be tested experimentally for specific cases.

With parallel tools, it may be a good idea to combine the results; for example, if we have three POS taggers, then whenever two agree, we can choose that option. There is still no simple way to calculate a combined accuracy, and no guarantee that the result will be an improvement. For some tools, this could also have unfortunate side effects. Suppose a parser is trying to determine the subject and object of a verb. A typical clause would have no more than one of each, so a parser might take that into account. But if we combine two parsers, we might increase the frequency of sentences analysed as having two subjects or two objects. This might even improve the per-word accuracy, but not the per-sentence accuracy.

Sometimes we have similar tools that give incompatible output; examples include POS taggers with different tagsets, and phrase vs. dependency parsers. In this case, we may still be able to use one to estimate the accuracy of the other, if their accuracies are known to correlate. There are also more complex ways to combine the outputs; taggers with different levels of specificity can be used to verify each other's results, etc.

4 Types of annotations in Språkbanken

4.1 Introduction

This is the main section of this report. It consists of descriptions of the different types of annotations that we use (or want to use) in Språkbanken. For each of the annotation types, the section tries to answer the following questions:

(1) What does annotation quality mean for this annotation type?

- What evaluation measures are there? (e.g., accuracy, precision, recall, f-score, LAS, UAS, BLEU, ...)
- What kind of confidence measures are there, that tools can return?
- What do these measures say? (both for evaluation and confidence)

(2) What annotation tools do we use currently?

- Are our current tools state-of-the-art?

- Are there any better alternatives that we should consider in the future?
- Do our current tools return any form of confidence score?

(3) What kind of annotated data is available?

- How is the data annotated?
- How can the data be used for Språkbanken's purposes?

Further reading

SIGANN: ACL Special Interest Group for Annotation
(https://www.cs.vassar.edu/sigann/)

LAW: Linguistic Annotation Workshop, 2007–2017
(https://aclanthology.coli.uni-saarland.de/sigs/sigann)

LAW+MWE 2018: http://aclweb.org/anthology/W18-49

4.2 OCR

Språkbanken has a large material of scanned old texts, primarily newspapers from the 19th century, called KubHist. This material is digitalised using OCR, but there is currently no reliable evaluation of the quality of the OCR. Adesam, Dannélls and Tahmasebi (2019) give an overview of some of the difficulties with OCR of old texts.

Evaluation measures for OCR The two primary evaluation measures for OCR are character error rate (CER) and word error rate (WER).

The project *En fri molntjänst för OCR* did an initial evaluation of OCR for Swedish blackletter texts printed between 1600 and 1800, and reported CER of 10–40% and a WER of 45–75% for two different off-the-shelf OCR tools (Borin, Bouma and Dannélls, 2016). The material and tools that have been developed in the project are available here: https://spraakbanken.gu.se/eng/ocr

4.3 Segmentation

Currently we are using the following automatic segmentations:

- Word segmentation / tokenisation
- Sentence segmentation
- Paragraph segmentation

The Sparv pipeline contains a number of different tokenisers adapted to produce different types of tokens (e.g., words, sentences) under different conditions (e.g., Old Swedish, lacking whitespaces after punctuation). In particular, segmentation can be performed based on line breaks, blank lines, punctuation, and paragraphs, plus CRF-based tokenisation for words intended specifically for Old Swedish. The main word and sentence tokenisers are based on NLTK (https://www.nltk.org) Punkt word and sentence tokenisers, respectively (Kiss and Strunk, 2006).

An alternative approach to the Punkt tokeniser that might be worth taking into account if it were decided to include new tokenisers is the one featured in the SpaCy toolkit (https://spacy.io). It implements its own word/punctuation tokenisation algorithm, and sentence segmentation is performed based on the dependency parse tree. A practical description of the process is given here: https://spacy.io/usage/linguistic-features#section-tokenisation. There is an ongoing effort to improve tokenisation for Swedish based on results obtained on the UD Swedish Talbank (Nivre and Megyesi, 2007); this is described here: https://github.com/explosion/spaCy/issues/2608.

Evaluation methods and measures A tokeniser can be evaluated on the basis of comparing its output to a chosen standard (such as a treebank). Given that there are different conventions that define finer points of tokenisation (e.g., should punctuation be dealt with as comprising tokens of its own? Should hyphenated words be split?), an evaluation standard should be chosen that fits the tokenising convention followed by the tokeniser of interest. The original evaluation of the Punkt tokeniser includes results on a Swedish corpus (composed mainly of news text from *Dagens nyheter*) from the European Corpus Initiative Multilingual Text (https://catalog.ldc.upenn.edu/LDC94T5).

A simple evaluation measure of a tokeniser is to obtain the distance between the tokeniser's output and the standard: in number of differing tokens or as Levenshtein distance (Dridan and Oepen, 2012; Grefenstette and Tapanainen, 1994). Such measures give a rough indication of the general performance of the tokeniser, but do not offer any insight into the types of errors it makes, and thus should be complemented with a qualitative inspection of said errors in order to determine the weaknesses of the system; an example of this type of qualitative evaluation is shown for different tokenisers applied to Swedish by Cap et al. (2016).

The most common difficulties for tokenisers stem from punctuation. While punctuation can be useful for determining sentence boundaries, it also signals abbreviations and other markers which might mislead tokenisers. If different uses of punctuation are labeled in the evaluation standard, the tokeniser's evaluation on those items can be defined as a classification task where the relevant items have to be assigned a class (e.g., sentence boundary, abbreviation, ellipsis). In such a case, evaluation measures typical of classification tasks can be used, such as precision, recall, and F-score (Kiss and Strunk, 2006; Read et al., 2012). This enables an easier distinction of the types of errors made by the tokeniser in cases related to punctuation, and is especially relevant for sentence tokenisation, since the ultimate goal is to correctly identify sentence

boundaries.

Further reading

Cap et al. (2016) "[...] we report on a pilot study of Swedish tokenization, where we compare the output of six different tokenizers on four different text types"

Grefenstette and Tapanainen (1994) "Here we will discuss tokenization as a problem for computational lexicography. [...] We present the roles of tokenization, methods of tokenizing, grammars for recognizing acronyms, abbreviations, and regular expressions such as numbers and dates"

Read et al. (2012) "We review the state of the art in automated sentence boundary detection (SBD) for English [...] We observe severe limitations in comparability and reproducibility of earlier work [...]"

Dridan and Oepen (2012) "We [...] present a new rule-based preprocessing toolkit that not only reproduces the Treebank tokenization with unmatched accuracy, but also maintains exact stand-off pointers to the original text and allows flexible configuration to diverse use cases [...] "

Nivre and Megyesi (2007) "[...] we describe an ongoing project with the aim of bootstrapping a large Swedish treebank [...] by reusing two previously existing annotated corpora [...]"

Kiss and Strunk (2006) "[...] we present a language-independent, unsupervised approach to sentence boundary detection. It is based on the assumption that a large number of ambiguities in the determination of sentence boundaries can be eliminated once abbreviations have been identified."

4.4 Lexical analysis

Currently Språkbanken is using analysers for the following lexical annotations:

- POS (part of speech)
- MSD (morpho-syntactic description)
- Lemma / lemgram / lexeme / sense
- Harmonisation / modernisation
- Multi-word units (multi-lexeme)
- Compound analysis

POS and MSD is currently annotated by the HunPos tagger trained on SUC (the Stockholm-Umeå corpus). In addition to the SUC tagset, we would like to include the universal postags, http://universaldependencies.org/u/pos/, and the Koala tags (Adesam, Bouma and Johansson, 2018). One point of using the Koala tagset is that it is developed as an

integrated whole together with the syntactic annotation, while the SUC tagset and the dependency annotations (see section 4.6) have been developed separately. The same argument goes for the universal postags and universal dependencies, which are being developed simultaneously.

Lemgrams are connecting text words (tokens) with entries in the lexicons (SALDO for modern Swedish, additionally Söderwall, Schlyter, Dalin, etc, for historical texts). Lemgrams point to a lexicon entry, as well as a part-of-speech and morphological paradigm. Lemmas are extracted from the lemgram-annotation. In addition, senses are listed and ranked using the WSD module described in section 4.8; given that there is a scored ranking of all possible senses (according to the inventory in SALDO) for a given lemma, this enables the possibility of informing the user about the sense annotation confidence.

Multi-word units are regular entries in e.g. SALDO. The Eukalyptus treebank also has specific multi-word annotation, which lies between the word and the syntactic level, using multi-word versions of parts-of-speech for labels and syntactic structure (phrases with secondary edges) for extent.

Compound analysis has been improved through the Koala project, now allowing for more parts than the previous two, and also ranking the compounds by probability.

Harmonisation of e.g. spelling variants is needed, both for historical texts, without a spelling norm, and for moderns texts, with spelling errors and web or SMS language spelling variants. One method for linking historical text variants to a historical lexicon is used in https://spraakbanken.gu.se/fsvreader/, see Adesam, Ahlberg and Bouma (2018).

Further reading

Östling (2018) "We perform a thorough PoS tagging evaluation on the Universal Dependencies treebanks, pitting a state-of-the-art neural network approach against UDPipe and our sparse structured perceptron-based tagger, EFSELAB. In terms of computational efficiency, EFSELAB is three orders of magnitude faster than the neural network model, while being more accurate than either of the other systems on 47 of 65 treebanks."

MWE Workshop on Multword Expressions, 2003–2018:

- http://multiword.sourceforge.net/PHITE.php?sitesig=CONF
- https://aclanthology.coli.uni-saarland.de/catalog?search_field=venue_name&q=MWE+workshop+multiword+expressions

4.5 Syntax: Overview

Currently the Språkbanken pipeline consists of the following syntactic analysers:

- Dependency parsing
- NER (named entity recognition)

We are not using the following:

- Shallow parsing (e.g., NP chunking)
- Phrase-structure (constituency) parsing
- Neural parsing models

Since many of the Språkbanken users would like to see phrases, and, as mentioned above, the Koala syntactic phrase annotation has been developed together with the part-of-speech tagset, this should be implemented in the future. In addition, using multiple types of syntactic annotation may be helpful in identifying annotation errors.

4.6 Syntax: Parsing

4.6.1 Available treebanks

The following Swedish treebanks are available. Some of them (e.g., Smultron or LinES) have very unstable URLs, that can be deprecated in any minute, so Språkbanken should offer to house the treebanks that are in the risk zone.

Talbanken (96k tokens, 6k sentences) was created in the 1970s and is manually annotated with lemmas, part-of-speech and syntactic structure. Talbanken is annotated with the following syntactic annotations:

- phrase-structure trees with functional labels on edges, according to the MAMBA annotation scheme (Teleman, 1974)
- MAMBA dependency trees (6316 sentences)
- Stanford dependency trees (6160 sentences)
- Universal dependency trees (6026 sentences) (UD Project, 2018)

All dependency tree annotations are automatically converted from the MAMBA syntactic annotation. These automatic conversions have not been evaluated, as far as I understand. More information about Talbanken:

- https://stp.lingfil.uu.se/~nivre/research/talbanken.html
- https://github.com/UniversalDependencies/UD_Swedish-Talbanken
- https://spraakbanken.gu.se/swe/resurs/talbanken

Talbanken05 (342k tokens, 21.5k sentences) As above, but contains all 4 original Talbanken sections with written and spoken data:

- https://stp.lingfil.uu.se/~nivre/research/Talbanken05.html

STB (1262k tokens, 80k sentences) Svensk Trädbank consists of two corpora:

- Talbanken (96k tokens, 6k sentences), see above
- Stockholm-Umeå Corpus (SUC, 1166k tokens, 74k sentences), manually annotated in the 1990s with part-of-speech and lemma

In order to build STB, Nivre et al. (2009):

1. automatically converted Talbanken's POS-tags to the SUC tagset
2. trained a phrase-structure parser on Talbanken, and ran than on the whole of SUC
3. manually corrected a gold-standard evaluation subset of 20k tokens (ca. 1.2k sentences) from SUC
4. automatically transformed the phrase-structure trees to dependency trees

So, STB consists of parse trees for 116k tokens (6k sentences) that have been manually corrected, and 1050k automatically annotated tokens (74k sentences). There are also possible conversion errors in steps (1) and (4), which as far as we understand have not been evaluated.

STB is annotated with the following syntactic annotations:

- phrase-structure trees in the same format as Talbanken
- MAMBA dependency trees

LinES (80k tokens, 4.6k sentences) The LinES Parallel Treebank consists of sentences translated from English, manually annotated with part-of-speech and UD dependency trees:

- https://www.ida.liu.se/~larah03/transmap/Corpus/
- https://github.com/UniversalDependencies/UD_Swedish-LinES

PUD (19k tokens, 1k sentences) The Swedish part of the Parallel Universal Treebank consists of sentences translated from English, manually annotated with part-of-speech and UD dependency trees:

- https://github.com/UniversalDependencies/UD_Swedish-PUD

Eukalyptus (100k tokens, 5.5k sentences) Treebank of modern Swedish:

- http://demo.spraakdata.gu.se/gerlof/Eukalyptus-0.1.0.zip

Smultron (35k tokens, 1.5k sentences) Stockholm MULtilingual TReebank is a parallel treebank with sentences in Swedish, English and German:

- https://www.cl.uzh.ch/en/texttechnologies/research/corpus-linguistics/paralleltreebanks/smultron.html

Syntag (100k tokens, 5.4k sentences) "SynTag is a so called tree bank, containing syntactically annotated text from 158 articles from the corpus Press-65, with about 100 000 running words. The annotation contains the relations of constituents and words, such as subjects or other arguments of finite verbs, in up to 12 levels of analysis. Additionally, there are simple word tags. The data still contains some errors, which will be corrected in the future."

- https://spraakbanken.gu.se/eng/resource/syntag

4.6.2 Tools for dependency parsing

The current Språkbanken parser is trained on Talbanken, using version 1.7.2 of the MaltParser. The model is available at the MaltParser webside:

- http://www.maltparser.org/mco/swedish_parser/swemalt.html

This model is probably not the best one possible, but we should retrain one (or several) new models on the treebanks mentioned in section 4.6.1.

The MaltParser system has also grown quite old and there are several more recent alternatives that are probably better:

- Stanford CoreNLP: https://stanfordnlp.github.io/CoreNLP/
- SpaCY: https://spacy.io/
- Apache OpenNLP: https://opennlp.apache.org/
- Allen NLP: https://allennlp.org/
- MaltParser: http://www.maltparser.org/
- UUParser, Uppsala: https://github.com/UppsalaNLP/uuparser/
- A parser for discontinuous phrase structures (Stanojevic and Alhama, 2017) has been adapted by Richard Johansson to output phrases with function labels: http://demo.spraakdata.gu.se/richard/withfunctions_t2_e.out.html

Future work There is recent work on non-projective dependency parsing, and graph-based (as opposed to tree-based) parsing, but we do not include any of those in this overview.

4.6.3 Evaluation measures for dependency parsing

The most common evaluation measures for dependency annotations are variants of attachment score – UAS (unlabeled) and LAS (labeled). LAS can be coarse- or fine-grained, if the functional labels in the annotation include features (such as number, gender, etc.). UAS/LAS are measured using accuracy, precision, recall and f-score. There are also variants to UAS/LAS, such as MLAS and BLEX which are used in the CoNLL 2018 shared task (see below).

Discussions about parser evaluation However, it is not at all clear if UAS/LAS are the best suited measures for Språkbanken's annotations. Here are some recent papers that discuss dependency evaluation in different ways:

- Tsarfaty, Nivre and Andersson (2011) discuss several methods for dependency evaluation, and proposes "a robust procedure for cross-experimental evaluation, based on deterministic unification-based operations for harmonizing different representations and a refined notion of tree edit distance for evaluating parse hypotheses relative to multiple gold standards"
- Nivre and Fang (2017) argue that "the usual attachment score metrics used to evaluate dependency parsers are biased in favor of analytic languages, where grammatical structure tends to be encoded in free morphemes (function words) rather than in bound morphemes (inflection). We therefore propose an alternative evaluation metric that excludes functional relations from the attachment score."
- Gulordava and Merlo (2016) propose "a method to evaluate the effects of word order of a language on dependency parsing performance, while controlling for confounding treebank properties. The method uses artificially-generated treebanks that are minimal permutations of actual treebanks with respect to two word order properties"
- Marvin and Linzen (2018) discuss a novel way of evaluating parsers; they "automatically construct a large number of minimally different pairs of English sentences, each consisting of a grammatical and an ungrammatical sentence. The sentence pairs represent different variations of structure-sensitive phenomena: subject-verb agreement, reflexive anaphora and negative polarity items. We expect a language model to assign a higher probability to the grammatical sentence than the ungrammatical one."
- There can very well be more relevant papers from IWPT, Int. Conf. on Parsing Technologies, e.g.: 15th IWPT, 2017, or 14th IWPT, 2015

Kübler, McDonald and Nivre (2009, section 6.1) give a description about the measures *Exact match*, *Attachment score*, and *Precision/recall*:

> "The standard methodology for evaluating dependency parsers, as well as other kinds of parsers, is to apply them to a test set taken from a tree-

bank and compare the output of the parser to the gold standard annotation found in the treebank. Dependency parsing has been evaluated with many different evaluation metrics. The most widely used metrics are listed here

- *Exact match*: This is the percentage of completely correctly parsed sentences. The same measure is also used for the evaluation of constituent parsers.
- *Attachment score*: This is the percentage of words that have the correct head. The use of a single accuracy metric is possible in dependency parsing thanks to the single-head property of dependency trees, which makes parsing resemble a tagging task, where every word is to be tagged with its correct head and dependency type. This is unlike the standard metrics in constituency-based parsing, which are based on precision and recall, since it is not possible to assume a one-to-one correspondence between constituents in the parser output and constituents in the treebank annotation.
- *Precision/Recall*: If we relax the single-head property or if we want to evaluate single dependency types, the following metrics can be used. They correspond more directly to the metrics used for constituent parsing.
  - Precision: This is the percentage of dependencies with a specific type in the parser output that were correct.
  - Recall: This is the percentage of dependencies with a specific type in the test set that were correctly parsed.
  - F-measure ($\beta = 1$): This is the harmonic mean of precision and recall.

All of these metrics can be unlabeled (only looking at heads) or labeled (looking at heads and labels). The most commonly used metrics are the labeled attachment score (LAS) and the unlabeled attachment score (UAS)."

Branch precision The text above is quoted in an answer at StackExchange, which also contains the following comment:

> "Another important metric for evaluating dependency parsing is the "branch precision". This metric measures the number of paths from root to nodes which contain only correctly annotated nodes. If the Root is identified incorrectly, the entire sentence is wrong. It considers the nodes closer to the root, in the dependency tree, more important, because they drive the logical structure of the sentence, the trunk."
>
> (https://linguistics.stackexchange.com/questions/6863/how-is-the-f1-score-computed-when-assessing-dependency-parsing)

Labelled and unlabelled attachment score Finally, the following is taken from the description of the CoNLL 2018 shared task:

> "The three main metrics are:
>
> - LAS (labeled attachment score) will be computed the same way as in the 2017 task so that results of the two tasks can be compared.
> - MLAS (morphology-aware labeled attachment score) is inspired by the CLAS metric computed in 2017, and extended with evaluation of POS tags and morphological features.
> - BLEX (bi-lexical dependency score) combines content-word relations with lemmatization (but not with tags and features)."
>
> (http://universaldependencies.org/conll18/evaluation.html)

Note that CoNLL 2018 goes from raw text to full annotation, which means that they have to handle segmentation differences. Here is how they do it:

> "The evaluation starts by aligning the system-produced words to the gold standard ones; a relation cannot be counted as correct if one of the connected nodes cannot be aligned to the corresponding gold-standard node. The aligning algorithm requires that the systems preserve the input sequence of non-whitespace characters. If a system uses a tokenizer or morphological analyzer that normalizes or otherwise damages the input characters, then the system must remember the original non-whitespace characters and restore them in a postprocessing step. Any multi-word tokens that the system produces must be properly marked as such, and the surface string to which they correspond must be indicated."

Unanswered questions We did not have time to investigate the following questions:

- Are there any existing parsers that give a confidence score for each dependency arc?
- If so, how can that score be incorporated into LAS, UAS, or any other quality measure?

4.6.4 Evaluation measures for constituency parsing

We did not have time to look into constituency parsing in this project, so this is left for future work. It is possible that many users would prefer to have annotation of phrases/constituents instead of dependencies.

According to Romanyshyn and Dyomkin (2014), there are (at least) the following evaluation metrics for syntactic parsing:

- Leaf-Ancestor Evaluation

- Parseval (several variants)
- Cross-bracketing
- Minimum Tree Edit Distance
- Transformations
- Error classification

4.7 Syntax: Named Entity Recognition (NER)

Existing NER tools Språkbanken's current system is originally by Dimitrios Kokkinakis, but then re-implemented by Jyrki Niemi (et al) in Helsinki who generalised it and converted to FSFT (Kokkinakis et al., 2014). It is a rule-based system, and cannot be trained or optimised by manually annotated data.

Unfortunately we didn't have much time to look into alternative tools for NER in this project, but EFSELAB (Östling, 2018) has apparently been used for training NER.

Evaluation measures Tha main evaluation measures are Precision/Recall, either Labeled or Unlabeled (i.e., if we take the type of entity into account). Here is a good overview blog post: http://www.davidsbatista.net/blog/2018/05/09/Named_Entity_Evaluation/

One tricky question is how to handle border-line cases. Here are some examples that can occur:

Gold standard	Annotation	Type of error
[A B] [C D]	[A B C D]	"merge"
[A B C D]	[A B] [C D]	"split"
[A B C D]	[A B C] D	"substring"
[A B C] D	[A B C D]	"superstring"
[A B C] D	A [B C D]	"overlap"

Note that the names of the error types are our own invention. These border-line cases are discussed by Dlugolinský, Ciglan and Laclavík (2013) (who differentiate between "lenient" and "strict" matching), and by Jiang, Banchs and Li (2016) (who talk about "partial boundary matching").

Further reading

NEWS More interesting papers can be found among the proceedings of NEWS, the Named Entity Workshop

6th NEWS 2016 https://aclanthology.coli.uni-saarland.de/volumes/proceedings-of-the-sixth-named-entity-workshop

7th NEWS 2018 https://aclanthology.coli.uni-saarland.de/volumes/proceedings-of-the-seventh-named-entities-workshop

Future/upcoming work

- Within Stian Rødven Eide's PhD project, the Riksdagen open data will be manually annotated för NER
- SWE-CLARIN has a working group for NER annotation (led by Lars Ahrenberg from Linköping University)

4.8 Semantics: Word sense disambiguation (WSD)

Sense annotation/disambiguation works by assigning a score to each possible sense of a word (according to the sense inventory of SALDO) and selecting the highest scoring sense. The scoring mechanism uses pre-trained word and word sense embeddings to perform a similarity computation using a dot product. The similarity is measured between each possible sense of a word (word sense embeddings) and the context formed by surrounding words (word embeddings) in a window of pre-defined size. The dot product between word sense embedding and (average of context) word embeddings determines how similar each sense is to the current context.

Alternatively to this approach, the WSD module provides two additional ones intended to provide baselines. (See Tools below.)

A slightly outdated but still very complete review of WSD, including approaches and evaluation methodology was compiled by Navigli (2009).

Resources The module depends on a Swedish sense inventory and sets of sense/word embeddings to work.

- *SALDO*: the SALDO lexicon provides the sense inventory (figures) that determines which senses are available when disambiguating any given word. This resource has a large influence on the problem, since essentially it defines the domain of the function mapping words to senses.
- *Embeddings*: The embeddings used are grouped in two sets: word sense embeddings which provide a representation for every word sense in the vocabulary, and word embeddings which provide a representation for each word in the vocabulary. For any disambiguation instance, sense vectors represent each possible sense of the target word being disambiguated and word vectors represent the context words in that instance.

  Given that this module produces probabilities for each sense, confidence levels can be easily integrated here based on those disambiguation probabilities. The list of these probabilities for polysemous words is currently part of the output shown in Sparv.

Tools The WSD module provides three approaches for disambiguation:

- *SCOUSE*: Our own method used by default, performs disambiguation by measuring the similarity of each word sense to the context provided by near-by words. This method is unsupervised (i.e., it is not trained on annotated corpora).
- *UKB*: A graph-based WSD approach that applies Personalized PageRank on the SALDO graph to disambiguate. It has good performance (comparable or superior to SCOUSE) but is rather slow in comparison, and hence it is not ideal for large amounts of text. The method is knowledge-based (i.e., it needs a knowledge database such as SALDO or WordNet to perform disambiguation).
- *First sense*: A baseline method which always choses the first sense in the order provided by SALDO (usually the most frequent one).

For a technical overview of the module, see the documentation here: https://github.com/spraakbanken/sparv-wsd/blob/master/README.pdf

Given the simplicity of the SCOUSE tool, it is probably not state-of-the-art (but it has not been measured against most existing systems). In any case, its performance depends heavily on the quality of the word and word sense embeddings used; e.g., it has been shown to perform both better and worse than UKB depending on the embeddings used. A recent review of the state-of-the-art systems was published by Le et al. (2018). Current best performing approaches use neural network-based WSD systems, but similar performance can be achieved training classifiers such as SVM.

Evaluation methods and measures A WSD task consists in selecting one (or more, if the tool/task allows it) out of several possible senses for an instance of an ambiguous word. Two variants can be distinguished: lexical sample, when the set of words to be disambiguated is restricted (e.g., one ambiguous word per sentence); and all-words, when all content words in a text have to be disambiguated. The annotation task in the pipeline is of the latter class.

A correctly disambiguated word should coincide with human intuition (as expressed in a gold standard, for example.) For any given polysemous word, one (or more) sense from an inventory is considered to be the correct "disambiguator". Accuracy (proportion of correctly disambiguated instances) or precision/recall/F1 are the typical measures used to evaluate the performance of an WSD system. This approach to evaluation relies thus on the quality of the annotated data used as gold standard; for this reason, when providing evaluation results, it would be informative to attach a measure of the annotation quality, such as inter-annotator agreement, if available.

Widely used benchmarks for English WSD include dedicated tasks from Senseval-1, 2, and 3, and Semeval-2007. All but Senseval-1 include all-words tasks besides lexical-sample tasks. Differences in sense inventories make some of these different benchmarks complementary among them. There is no Swedish benchmark for WSD, but given the development of the sense-annotated Koala, there is an opportunity to develop

one with a moderate amount of effort. Excerpts from Koala have been used to evaluate WSD systems by Nieto Piña and Johansson (2016; 2017).

Annotation quality In order to provide a measure of the annotation quality provided by the WSD module, the WSD tool's own confidence measure could be used. As explained above, the tool works by scoring each possible sense for an instance of a given word; this score is normalised to the interval [0,1] and is interpreted as a probability. The most probable sense is the one chosen to annotate an instance, but each possible sense has its own probability. The list of all senses and their probabilities is currently shown in Sparv. The interpretation of this probability score is fairly straightforward: given a word w with possible senses s_1 and s_2, any disambiguation that assigns a probability > 0.5 to s_2 will be correct, but the confidence with a probability of 0.9 for s_2 is greater than if it were 0.55.

Further reading

Antoine, Villaneau and Lefeuvre (2014) "This paper presents an experimental study with four measures (Cohen's κ, Scott's π, binary and weighted Krippendorff's α) on three tasks: emotion, opinion and coreference annotation."

Navigli (2009) "In this paper, we have gone through a survey regarding the different approaches adopted in different research works, the State of the Art in the performance in this domain, recent works in different Indian languages and finally a survey in Bengali language."

Le et al. (2018) "the technique proposed by Yuan et al. (2016) returned state-of-the-art performance in several benchmarks, but neither the training data nor the source code was released. This paper presents the results of a reproduction study and analysis of this technique using only openly available datasets (GigaWord, SemCor, OMSTI) and software (TensorFlow)."

Raganato, Camacho-Collados and Navigli (2017) "In this paper we develop a unified evaluation framework and analyze the performance of various Word Sense Disambiguation systems in a fair setup. The results show that supervised systems clearly outperform knowledge-based models."

Camacho-Collados and Pilehvar (2018) "This survey focuses on the representation of meaning. [...] We present a comprehensive overview of the wide range of techniques in the two main branches of sense representation, i.e., unsupervised and knowledge-based."

Borin, Forsberg and Lönngren (2013) "The resource described here—SALDO—is such a lexical–semantic resource, intended primarily for use in language technology applications, and offering an alternative organization to PWN-style wordnets."

Pilán (2015) "This paper describes a knowledge-based approach to word-sense disambiguation using a lexical-semantic resource, SALDO. [...] The proposed method is based

on maximizing the overlap between associated word senses of nouns and verbs co-occuring within a sentence."

Johansson and Nieto Piña (2015) "We present a framework for using continuous-space vector representations of word meaning to derive new vectors representing the meaning of senses listed in a semantic network. [...] It uses two ideas: first, that vectors for polysemous words can be decomposed into a convex combination of sense vectors; secondly, that the vector for a sense is kept similar to those of its neighbors in the network. "

Nieto Piña and Johansson (2016) "We propose a simple graph-based method for word sense disambiguation (WSD) where sense and context embeddings are constructed by applying the Skip-gram method to random walks over the sense graph."

Nieto Piña and Johansson (2017) "We propose to improve word sense embeddings by enriching an automatic corpus-based method with lexicographic data. [...] The incorporation of lexicographic data yields embeddings that are able to reflect expert-defined word senses, while retaining the robustness, high quality, and coverage of automatic corpus-based methods. "

4.9 Semantics: Sentiment Analysis

Sentiment annotation in the Sparv pipeline is performed at the word sense level, where words are assigned a tag (*positive*, *neutral*, or *negative*) based on a word sense sentiment lexicon created by the underlying model; the sentiment lexicon has been manually curated. The labels are assigned according to a score which is also present in the annotation output from the pipeline.

Resources SenSALDO sentiment lexicon: provides a list of SALDO entries with their assigned sentiment score, which is used to derive the sentiment label.

Model The SenSALDO lexicon is created by a support vector classifier applied to word sense embeddings derived from Johansson and Nieto Piña (2015). This model assigns a sentiment score in $[-1, 1]$ to each sense, from which the tags originate by rounding the score to the closest integer. The lexicon is partially manually curated after senses have been classified. (See Rouces et al. (2018a) for details.)

Since the model relies on word sense embeddings, its accuracy depends partly on the sense embedding model used. Furthermore, words with multiple senses are assigned a sentiment tag based on a weighted average of the sentiment score for each sense, where the weight is given by the WSD module.

This model is the best-performing one of a number of approaches explored that include graph-based methods besides embedding-based ones (Rouces et al., 2018b).

Evaluation methods and measures Sentiment analysis at the word level, as a classification task, can be evaluated using standard measures like accuracy or precision / recall / F-score based on a gold standard that can be a manually crafted sentiment lexicon; these measures indicate the ability of the annotation system to tag the sentiment of a word according to the annotation contained in the gold standard. This is the approach used to evaluate SenSALDO (Rouces et al., 2018a), which also contains additional measures like Kendall tau and Spearman correlation for completeness. The results presented in that work show that the approach implemented in the pipeline currently has the best performance of all systems compared in the study.

Current best-performing sentiment analysis methods tend to use word representations like embeddings as features for a classifier (see a recent comparison of state-of-the-art models in Barnes, Klinger and im Walde, 2017). However, sentiment analysis is usually focused on predicting the sentiment of sentences or paragraphs (e.g., reviews or tweets), while word-level sentiment is usually considered an intermediate step towards predicting the sentiment of those larger units. As such, evaluation of this model on standard sentiment benchmarks (e.g., OpenNer, Semeval 2013 Twitter dataset, Stanford Sentiment) is not applicable. Furthermore, these benchmarks tend to be focused on the English language. Since the amount of research in sentiment analysis for Swedish is rather limited, any future efforts in comprehensive evaluation of this tool would likely involve defining an evaluation framework; this could be an opportunity to establish evaluation standards for Swedish sentiment analysis.

Annotation confidence The word-based sentiment annotation tool returns a score that could be interpreted in terms of a confidence score: The sentiment tag (positive, neutral, negative) of any annotated word is based on a sentiment score on the range $[-1, 1]$. Values of this score close to the extremes indicate a high confidence of the tool in labeling a word as positive or negative, while middle values are labeled as neutral. Values close to the cut-off scores that separate positive/negative tags from neutral would indicate a lower confidence on the sentiment of the tagged word, since the cut-off values are arbitrary. An interpretation of this score could thus be used to inform the annotation confidence.

Further reading

Rouces et al. (2018a) "This paper describes the development of an extensive sentiment lexicon for written (standard) Swedish. [...] We implement and evaluate three methods: a graph-based method that iterates over the SALDO structure, a method based on random paths over the SALDO structure and a corpus-driven method based on word embeddings."

Rouces et al. (2018b) "In this paper we describe the creation of SenSALDO, a comprehensive sentiment lexicon for Swedish [...]"

Barnes, Klinger and im Walde (2017) "In this paper, we [...] comparing several models on six different benchmarks, which belong to different domains and additionally

have different levels of granularity [...]"

http://alt.qcri.org/semeval2017/task4/ SemEval 2017, task 4 "Sentiment Analysis in Twitter", contains a list of different evaluation measures, depending on the classification task:

- positive–neutral–negative: the task uses "macroaveraged recall (recall averaged across the three classes)," which has "better theoretical properties" than normal P/R/F1
- five-point scale: the task uses "macroaveraged mean absolute error," but they also added a secondary measure, "an extension of macroaveraged recall for ordinal regression"

4.10 Annotations that we do not cover

Text-level attributes One kind of text-level annotation that we have is *readability metrics*, which we use in our learner corpora. Other text-level annotations that currently are not included in Språkbanken are: *simplification*, *summarisation*, and *categorisation* at the text level.

Regarding these, Sulem, Abend and Rappoport (2018) argue that BLEU is not a good evaluation metrics for text simplification, and ShafieiBavani et al. (2018) propose an improvement of the ROGUE metric for text summarisation.

Parallel alignment The main question here is what tools we use for aligning our parallel corpora, and how they can be evaluated and improved. We did not have the time to look into this.

FrameNet We have not looked into FrameNet and in what way FrameNet is / can be used in Språkbanken's annotations.

Bring conceptual classes We have not looked into the conceptual classes that Bring (1930) introduced.

5 Text similarity

Assume that we have a POS tagger trained and evaluated on the SUC corpus. We know that the accuracy on SUC is 97%. How can we estimate its accuracy on another corpus? Since SUC contains newspaper texts, it should work better on GP/DN than on blog texts, but can we get a better idea of the accuracy?

To answer this, we need some kind of estimation of the similarity between different text types. In some cases, we may be able to base this on metadata about the type of text, but for wider applicability we may need to rely on textual data. We have not been able to find any previous studies using this approach for accuracy estimation.

Similarity as a field is closely related to classification, and has been studied for many other purposes. We would have to select on one hand which features to extract from the text, and on the other hand which mathematical method to apply to them.

As for features, it seems reasonable, if not strictly necessary, to base them on data held before the step being analysed, so that for example we would not use data from parsing in our analysis of POS tagging. The most simple approach would be to use data found directly in the text, such as frequencies of words, word n-grams, or character n-grams. Judging from previous research, we would suspect that the difference between those is insignificant, and that nothing is lost by not using later-stage features, but it may be worth testing nonetheless.

As for mathematical measures of similarity, there are a few to choose from, such as cosine similarity or cartesian distance. Here, too, we suspect that they would give very similar results, but we may want to verify that.

Preliminary tests show that similarity scores (based on word frequencies and cosine similarity) can in principle be used to predict accuracy, but the reliability is disappointingly low, so the method may have limited practical use.

This could be seen as a failure of this similarity measure, but it could also be taken to show that the accuracies of new texts are just difficult to predict. A possible experiment here would be to check how consistent the accuracies are, by checking accuracies for parts of the subcorpora and calculating the correlation.

One thing that is clear from SUC and Eukalyptus is that the cross-subcorpus accuracy for POS tagging is considerably lower. While the same-subcorpus accuracy (training on one half, testing on the other) is typically around 99%, the cross-subcorpus accuracy is mainly in the 90–95% range for SUC, and 83–93% for Eukalyptus. This in itself is of course useful information on the accuracy for dissimilar texts.

6 Manual annotation

The main question for this section is how we can incorporate manual annotations in our quality assessments. This problem has not been addressed in this project, but we are listing some relevant references for future work.

In the future we might want to allow users to manually check/correct our annotations. This will lead to an automatically annotated corpus with some manual corrections. How will this affect the evaluation/confidence of our annotation tools?

6.1 Pure manual annotations

This is only relevant for new annotation projects, which occur relatively infrequently. There are only a few manually annotated corpora in Språkbanken, e.g., SUC, Eukalyptus, Talbanken, etc.

Some questions to think about if/when we start a new annotation project:

- how does IAA (inter-annotator agreement) affect the final quality of the tools that are trained on the data?
- can we get a confidence measure even if we only have one annotator?

Further reading

Passonneau and Carpenter (2013) "This paper presents a case study of a difficult and important categorical annotation task (word sense) to demonstrate a probabilistic annotation model applied to crowdsourced data."

Dickinson (2015) "This paper surveys methods for annotation error detection and correction. Methods can broadly be characterized as to whether they detect inconsistencies with respect to some statistical model based only on the corpus data or whether they detect inconsistencies with respect to a grammatical model, in general, some external information source."

Sperber et al. (2016) "In this paper, we attempt to bridge this gap by proposing a framework for lightly supervised quality estimation. We collect manually annotated scores for a small number of segments in a test corpus or document, and combine them with automatically predicted quality scores for the remaining segments to predict an overall quality estimate."

6.2 Mixed manual / automatic annotations

"[T]he term *silver standard* describes a level of annotation quality between a manually created gold standard and the unchecked output of automatic processing" (Eckart and Gärtner, 2016). Because of the costs involved in building manually annotated corpora from scratch, silver-standards could be a feasible alternative. One Swedish example of a silver-standard corpus is STB (Svensk Trädbank), where the morphological and syntactic annotations were checked and revised semi-automatically.

Further reading

Nivre and Megyesi (2007) "[. . .] we describe an ongoing project with the aim of bootstrapping a large Swedish treebank [. . .] by reusing two previously existing annotated corpora [. . .]"

https://stp.lingfil.uu.se/~nivre/swedish_treebank "This annotation has been projected to SUC by training a parser on Talbanken, parsing the entire SUC corpus, and manually revising a small sample of about 20,000 tokens to be used for evaluation purposes, which we will refer to as the gold standard section of SUC. Sentences that have not been revised manually have been flagged automatically in case they contain configurations of structural and functional categories that are not licensed by the annotation scheme. [...] Finally, we have automatically converted the constituent structure annotation in both Talbanken and SUC, using head percolation rules to determine the head of each phrase and using a subset of the grammatical functions to label dependency edges."

Rebholz-Schuhmann et al. (2010) "The CALBC initiative aims to provide a large-scale biomedical text corpus that contains semantic annotations for named entities of different kinds. [...] During the harmonization phase, the results produced from those different systems were integrated in a single harmonized corpus ("silver standard" corpus) by applying a voting scheme."

Hahn et al. (2010) "[T]he current construction policy for such a silver standard requires crucial parameters [...] to be set a priori, based on extensive testing though, at corpus compile time. Accordingly, such a corpus is static, once it is released. We here propose an alternative policy where silver standards can be dynamically optimized and customized on demand (given a specific goal function) using a gold standard as an oracle."

Filannino and Bari (2015) "Collecting and manually annotating gold standards in NLP has become so expensive that in the last years the question of whether we can satisfactorily replace them with automatically annotated data (silver standards) is arising more and more interest. We focus on the case of dependency parsing for Italian and we investigate whether such strategy is convenient and to what extent. Our experiments, conducted on very large sizes of silver data, show that quantity does not win over quality."

Eckart and Gärtner (2016) "We present our approach for annotating a large collection of non-standard multimodal data. Its automatically created silver standard annotations lack the quality of their manual counterparts but will be enriched with confidence estimations which allow an assessment of an annotation's expected correctness."

6.3 Finding sentences to annotate / check manually

Assume we want to (*a*) evaluate the automatic annotation, and/or (*b*) improve the accuracy of the tool by retraining. Are there automatic ways for finding good candidates, such that the confidence/accuracy increases as much as possible with the least effort?

Further reading

Dickinson (2015) See section 6.1 for abstract.

Sagot and de la Clergerie (2006) "We introduce an error mining technique for automatically detecting errors in resources that are used in parsing systems. We applied this technique on parsing results produced on several million words by two distinct parsing systems, which share the syntactic lexicon and the pre-parsing processing chain. We were thus able to identify missing and erroneous information in these resources."

Xu, Yang and Huang (2016) "[...] annotation of the full corpus sentence by sentence is resource intensive. In this paper, we propose an approach that iteratively extracts frequent parts of sentences for annotating, and compresses the set of sentences after each round of annotation. Our approach can also be used in preparing training sentences for binary classification [...]"

References

Adesam, Yvonne. 2012. The multilingual forest: Investigating high-quality parallel corpus development. Ph.D. diss., Stockholm University, Stockholm, Sweden. https://www.semanticscholar.org/paper/The-Multilingual-Forest---Investigating-Parallel-Adesam/77c2ccbfc550a149d576700469b3f417ef2278ea.

Adesam, Yvonne, Malin Ahlberg and Gerlof Bouma. 2018. FSvReader – exploring Old Swedish cultural heritage texts. Tolonen Jouni Tuominen Eetu, Mäkelä Mikko (ed.), *3rd conference of the digital humanities in the Nordic countries*, Volume 2084 of *CEUR Workshop Proceedings*. University of Helsinki, Faculty of Arts. http://ceur-ws.org/Vol-2084/shortplus6.pdf.

Adesam, Yvonne, Gerlof Bouma and Richard Johansson. 2018. The Koala part-of-speech and morphological tagset for Swedish. *SLTC 2018*. https://gup.ub.gu.se/publication/273841.

Adesam, Yvonne, Dana Dannélls and Nina Tahmasebi. 2019. Exploring the quality of the digital historical newspaper archive KubHist. In preparation.

Antoine, Jean-Yves, Jeanne Villaneau and Anaïs Lefeuvre. 2014. Weighted Krippendorff's alpha is a more reliable metrics for multi-coders ordinal annotations: experimental studies on emotion, opinion and coreference annotation. *EACL 2014*. http://aclweb.org/anthology/E14-1058.

Barnes, Jeremy, Roman Klinger and Sabine Schulte im Walde. 2017. Assessing state-of-the-art sentiment models on state-of-the-art sentiment datasets. *EMNLP 2017*. http://aclweb.org/anthology/W17-5202.

Borin, Lars, Gerlof Bouma and Dana Dannélls. 2016. A free cloud service for OCR. Technical Report GU-ISS-2016-01, University of Gothenburg. https://gupea.ub.gu.se/handle/2077/42228.

Borin, Lars, Markus Forsberg and Lennart Lönngren. 2013. SALDO: a touch of yin to WordNet's yang. *Language Resources and Evaluation* 47 (4): 1191–1211. https://link.springer.com/article/10.1007/s10579-013-9233-4.

Bring, Sven Casper. 1930. *Svenskt ordförråd ordnat i begreppsklasser*. Stockholm: Hugo Gebers förlag. http://runeberg.org/bring/.

Camacho-Collados, Jose and Mohammad Taher Pilehvar. 2018. From word to sense embeddings: A survey on vector representations of meaning. *Journal of Artificial Intelligence Research* 63: 743–788. https://arxiv.org/pdf/1805.04032.pdf.

Cap, Fabienne, Yvonne Adesam, Lars Ahrenberg, Lars Borin, Gerlof Bouma, Markus Forsberg, Viggo Kann, Robert Östling, Aaron Smith, Mats Wirén and Joakim Nivre. 2016. SWORD: Towards cutting-edge Swedish word processing. *SLTC 2016*. Umeå, Sweden. http://www8.cs.umu.se/~johanna/sltc2016/abstracts/SLTC_2016_paper_13.pdf.

Dickinson, Markus. 2015. Detection of annotation errors in corpora. *Language and Linguistics Compass* 9 (3): 119–138. https://onlinelibrary.wiley.com/doi/full/10.1111/lnc3.12129.

Dlugolinský, Štefan, Marek Ciglan and Michal Laclavík. 2013. Evaluation of named entity recognition tools on microposts. *17th international conference on intelligent engineering systems (INES)*. San Jose, Costa Rica. https://ieeexplore.ieee.org/abstract/document/6632810.

Dridan, Rebecca and Stephan Oepen. 2012. Tokenization: Returning to a long solved problem – A survey, contrastive experiment, recommendations, and toolkit –. *Proceedings of ACL 2012 (volume 2: short papers)*, 378–382. http://aclweb.org/anthology/P12-2074.

Eckart, Kerstin and Markus Gärtner. 2016. Creating silver standard annotations for a corpus of non-standard data. *13th conference on natural language processing (KONVENS 2016)*. Bochum, Germany. https://www.linguistics.rub.de/konvens16/pub/12_konvensproc.pdf.

Filannino, Michele and Marilena Di Bari. 2015. Gold standard vs. silver standard: the case of dependency parsing for Italian. *Second Italian conference on computational linguistics CLiC-it 2015*. Trento, Italy. http://books.openedition.org/aaccademia/pdf/1475.

Grefenstette, Gregory and Pasi Tapanainen. 1994. What is a word, what is a sentence? Problems of tokenization. *Proceedings of COMPLEX'94*. Budapest. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.28.5162.

Gulordava, Kristina and Paola Merlo. 2016. Multi-lingual dependency parsing evaluation: a large-scale analysis of word order properties using artificial data. *TACL* 4: 343–356. http://aclweb.org/anthology/Q16-1025.

Hahn, Udo, Katrin Tomanek, Elena Beisswanger and Erik Faessler. 2010. A proposal for a configurable silver standard. *Fourth linguistic annotation workshop, ACL 2010*. Uppsala, Sweden. http://www.aclweb.org/anthology/W10-1838.

Jiang, Ridong, Rafael E. Banchs and Haizhou Li. 2016. Evaluating and combining named entity recognition systems. *6th named entity workshop*. Berlin, Germany. http://www.aclweb.org/anthology/W16-2703.

Johansson, Richard and Luis Nieto Piña. 2015. Embedding a semantic network in a word space. *NAACL-2015 conference of the north American chapter of the association for computational linguistics: Human language technologies*, 1428–1433. http://www.aclweb.org/anthology/N15-1164.

Kiss, Tibor and Jan Strunk. 2006. Unsupervised multilingual sentence boundary detection. *Computational Linguistics* 32 (4): 485–525. https://www.mitpressjournals.org/doi/pdfplus/10.1162/coli.2006.32.4.485.

Kokkinakis, Dimitrios, Jyrki Niemi, Sam Hardwick, Krister Lindén and Lars Borin. 2014. HFST-SweNER: A new NER resource for Swedish. *Proceedings of the 9th edition of the language resources and evaluation conference (LREC)*. Reykjavik, Iceland. http://www.lrec-conf.org/proceedings/lrec2014/summaries/391.html.

Kübler, Sandra, Ryan McDonald and Joakim Nivre. 2009. *Dependency parsing*. Volume 1.1 of *Synthesis Lectures on Human Language Technologies*. Morgan and Claypool. https://doi.org/10.2200/S00169ED1V01Y200901HLT002.

Le, Minh, Marten Postma, Jacopo Urbani and Piek Vossen. 2018. A deep dive into word sense disambiguation with LSTM. *Proceedings of the 27th international conference on computational linguistics*, 354–365. http://www.aclweb.org/anthology/C18-1030.

Marvin, Rebecca and Tal Linzen. 2018. Targeted syntactic evaluation of language models. *EMNLP-18*. http://aclweb.org/anthology/D18-1151.

Navigli, Roberto. 2009. Word sense disambiguation: A survey. *ACM computing surveys (CSUR)* 41 (2): 10. https://arxiv.org/pdf/1508.01346.pdf.

Nieto Piña, Luis and Richard Johansson. 2016. Embedding senses for efficient graph-based word sense disambiguation. *Proceedings of TextGraphs-10: the workshop on graph-based methods for natural language processing*, 1–5. http://www.aclweb.org/anthology/W16-1401.

Nieto Piña, Luis and Richard Johansson. 2017. Training word sense embeddings with lexicon-based regularization. *Proceedings of the eighth international joint conference on natural language processing (volume 1: long papers)*, Volume 1, 284–294. http://www.aclweb.org/anthology/I17-1029.

Nivre, Joakim and Chiao-Ting Fang. 2017. Universal dependency evaluation. *Nodalida 2017 workshop on universal dependencies (UDW-17)*. http://aclweb.org/anthology/W17-0411.

Nivre, Joakim and Beata Megyesi. 2007. Bootstrapping a Swedish treebank using cross-corpus harmonization and annotation projection. *Proceedings of the 6th international workshop on treebanks and linguistic theories*, 97–102. http://tlt07.uib.no/papers/11.pdf.

Nivre, Joakim, Beáta Megyesi, Sofia Gustafson-Capková, Filip Salomonsson, Bengt Dahlqvist, Anna Sågvall Hein, Johan Hall and Jens Nilsson. 2009. Svensk trädbank. https://stp.lingfil.uu.se/~nivre/swedish_treebank/.

Passonneau, Rebecca J. and Bob Carpenter. 2013. The benefits of a model of annotation. *LAW7, 7th linguistic annotation workshop and interoperability with discourse*. Sofia, Bulgaria. http://www.aclweb.org/anthology/W13-2323.

Pilán, Ildikó. 2015. Helping Swedish words come to their senses: word-sense disambiguation based on sense associations from the SALDO lexicon. *Proceedings of the 20th nordic conference of computational linguistics (NODALIDA 2015)*, 275–279. http://www.aclweb.org/anthology/W15-1836.

Raganato, Alessandro, Jose Camacho-Collados and Roberto Navigli. 2017. Word sense disambiguation: A unified evaluation framework and empirical comparison. *Proceedings of the 15th conference of the European chapter of the ACL: Volume 1, long papers*, Volume 1, 99–110. http://www.aclweb.org/anthology/E17-1010.

Read, Jonathon, Rebecca Dridan, Stephan Oepen and Lars Jørgen Solberg. 2012. Sentence boundary detection: A long solved problem? *Proceedings of COLING 2012: Posters*, 985–994. Mumbai: ACL. http://aclweb.org/anthology/C12-2096.

Rebholz-Schuhmann, Dietrich, Antonio Jose Jimeno Yepes, Erik M Van Mulligen, Ning Kang, Jan Kors, David Milward, Peter Corbett, Ekaterina Buyko, Elena Beisswanger and Udo Hahn. 2010. CALBC silver standard corpus. *Journal of bioinformatics and computational biology* 8 (1): 163–179. https://www.ncbi.nlm.nih.gov/pubmed/20183881.

Romanyshyn, Mariana and Vsevolod Dyomkin. 2014. The dirty little secret ofcConstituency parser evaluation. https://tech.grammarly.com/blog/the-dirty-little-secret-of-constituency-parser-evaluation.

Rouces, Jacobo, Nina Tahmasebi, Lars Borin and Stian Rødven Eide. 2018a. SenSALDO: Creating a sentiment lexicon for Swedish. *Proceedings of the eleventh international conference on language resources and evaluation (LREC-2018)*. http://www.aclweb.org/anthology/L18-1662.

Rouces, Jacobo, Nina Tahmasebi, Lars Borin and Stian Rødven Eide. 2018b. SenSALDO: a Swedish sentiment lexicon for the SWE-CLARIN toolbox. *CLARIN annual conference 2018*, 173. https://ris.utwente.nl/ws/portalfiles/portal/63914792/CE_2018_1292_CLARIN2018_ConferenceProceedings.pdf#page=180.

Sagot, Benoît and Éric de la Clergerie. 2006. Error mining in parsing results. *COLING-ACL*. Sydney, Australia. http://aclweb.org/anthology/P06-1042.

ShafieiBavani, Elaheh, Mohammad Ebrahimi, Raymond Wong and Fang Chen. 2018. A graph-theoretic summary evaluation for ROUGE. *EMNLP 2018, conference on empirical methods in natural language processing*. Brussels, Belgium. http://aclweb.org/anthology/D18-1085.

Sperber, Matthias, Graham Neubig, Jan Niehues, Sebastian Stüker and Alex Waibel. 2016. Lightly supervised quality estimation. *COLING 2016, the 26th international conference on computational linguistics*. Osaka, Japan. http://aclweb.org/anthology/C16-1292.

Stanojevic, Miloš and Raquel G. Alhama. 2017. Neural discontinuous constituency parsing. *Proceedings of EMNLP, empirical methods in natural language processing*. Copenhagen, Denmark: ACL. http://aclweb.org/anthology/D17-1174.

Sulem, Elior, Omri Abend and Ari Rappoport. 2018. BLEU is not suitable for the evaluation of text simplification. *EMNLP-2018*. Brussels, Belgium. http://aclweb.org/anthology/D18-1081.

Teleman, Ulf. 1974. *Manual för grammatisk beskrivning av talad och skriven svenska*. Lund: Studentlitteratur.

Tsarfaty, Reut, Joakim Nivre and Evelina Andersson. 2011. Evaluating dependency parsing: Robust and heuristics-free cross-annotation evaluation. *EMNLP 2011*. http://aclweb.org/anthology/D11-1036.

UD Project. 2018. Universal Dependencies. http://universaldependencies.org.

Xu, Ge, Xiaoyan Yang and Chu-Ren Huang. 2016. Selective annotation of sentence parts: Identification of relevant sub-sentential units. *12th workshop on Asian language resources*. Osaka, Japan. http://www.aclweb.org/anthology/W16-5411.

Östling, Robert. 2018. Part of speech tagging: Shallow or deep learning? *North European Journal of Language Technology* 5: 1–15. http://dx.doi.org/10.3384/nejlt.2000-1533.1851.

GU-ISS, Forskningsrapporter från Institutionen för svenska språket, är en oregelbundet utkommande serie, som i enkel form möjliggör spridning av institutionens skriftliga produktion. Det främsta syftet med serien är att fungera som en kanal för preliminära texter som kan bearbetas vidare för en slutgiltig publicering. Varje enskild författare ansvarar för sitt bidrag.

GU-ISS, Research reports from the Department of Swedish, is an irregular report series intended as a rapid preliminary publication forum for research results which may later be published in fuller form elsewhere. The sole responsibility for the content and form of each text rests with its author.

Forskningsrapporter från institutionen för svenska språket, Göteborgs universitet
Research Reports from the Department of Swedish

ISSN 1401-5919

www.svenska.gu.se/publikationer/GU-ISS