Mapping Inherit sentential logic from argumentative text using cue words

Alexander Wallin
LTH

Christian Lindgren
LTH

alexander@threadsafe.se christian.lindgren@lus.lu.se

Abstract

In this project we assume that an argumentative text is constructed from individually evaluable constituents spanning up an argument, a format with certain structural similarities with sentential logic.

This project demonstrates to some extent there is a possibility to tag natural language sentences with sentential logical rules for the swedish language. By using cutting edge tools for processing of swedish language and text from swedish Wikipedia(a) an application for automatic extraction of sentential logic rules from complete sentences were created. The methodology allows for any POS-tagger and dependency parser to be used, however our implementation is heavily dependant on Stagger(b) and MaltParser(c).

Key words: NLP, Sentential Logic, Swedish, Dependency Parsing, CoNLL, Implementation

1 Introduction

The purpose of this project was the extraction of sentential logical rules from argumentative text.

Sentential logic is inferring conclusions from a given set of logical premises. A logic expression can in its simplest form be a single statement letter. This simple form generally consists of an assignment of a property onto an object (i.e. "the sky is blue"), which ultimately can be evaluated as either true or false. Furthermore there are connectives taking one or more statement letters and evaluates them onto a new statement letter. An expression will span a tree using connectives as its inner nodes and statement letters as its leafs, leaving the root node evaluable as true or false thus validating the correctness of an expression. (Andersson, 2012, p.4)

To proceed with extracting the counterparts to connectives and statement letters from an argumentative text the path through a dependency structured text was assumed to be a good option, as linguistic theory allows for models with inherent logical structures. In order of creating a dependency representation the text first has to be part-of-speech tagged (POS-tagged).

POS-tagging annotates each word in a sentence with its predicted part of speech. The tagging system used here is called Stagger, by Robert Östling, which has been made freely available. (Östling, 2012)

A dependency structure consists of a tree where each word in a sentence is represented as a node and has a single ingoing edge tagged with a dependency type (functional category). (Hall et al., 2007, p.284-285) The dependency parser used here is called MaltParser, by Johan Hall, Jens Nilsson and Joakim Nivre.

The raw data used for processing has been extracted from the swedish Wikipedia.

The ConLL data format is a standard tab delimited format used for Natural Language Parsing which undergoes changes in accordance with the needs of the yearly Conference on Computational Natural Language Learning. Because of this and the tools used the format used for this project was based upon ConLL 2006.(Buchholz and Marsi, 2006)

The evaluation of our methodology was made by the creation of an application that applies our constructed logical rules on swedish sentences and visualizes them with a modified tree graph based upon the dependency parsed sentence content.

2 Method

2.1 Annotating the text

In order of annotating any unannotated text the same procedure of annotation has been used. In the

case of POS-tagging the precreated model available on Stagger webpage has been used. The model has been trained using a corpus, SUC 3.0 (Stockholm-Umeå Corpus), and a lexicon, SALDO. The model is expected to yield a correctness of 92 % on user-generated content. (Östling, 2012) Regarding the dependency parsing MaltParser provides a model as well via their webpage. This model has been trained using the Swedish Treebank corpus.

2.2 Creating sentential logic candidates

A list of possible candidate cue words were generated from the writer's inherit knowledge of the language and were ordered according to presumed complexity. The words chosen were then mapped by hand to the presumed sentential logic form. The words corresponding to the most important sentential logic form were then chosen at the basis for further analysis. (Andersson, 2012, p.4-5)

2.3 Candidate analysis and rule extraction

Sentences containing the candidate words were then extracted from wikipedia and their interdependency with the parsed dependency tree were enumerated and mapped according to ingoing and outgoing functional categories. The largest consistent group were then checked by hand to ensure that most were correctly mappable to sentential logic.

2.4 Candidate validation and rule implementation

The largest consistent group were then checked by hand to ensure that most sentences would be correctly mappable to sentential logic and if such would be the case then rules would be created according to the accrued parameters.

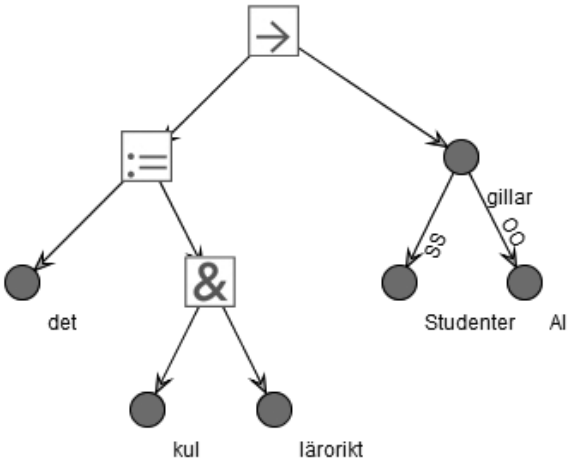
3 Results

Word(s)	Connectives	Occurences
och	&	1 476 511
eller	—	81 614
därför att	→	3040
ger	→	20 647
är	:=	401 936
inte	!	369 156

Tabell 1: Cue word extraction

Word(s)	Coverage	False Positives
och	61%	60%
eller	81%	5%
därför att	46%	50%
ger	4%	30%
är	30%	10%
inte	100%	80%

Tabell 2: Cue word statistics



Figur 1: Application output example

The result in 3 was extracted and parsed from the sentence "Studenter gillar AI därför att det är kul och lärorikt" which correctly extracts the inherit logical relationships based upon previously extracted cue word rules.

4 Analysis

4.1 Word ambiguity

Certain words are ambiguous by nature with different meaning, such as "därför att" that may mean either "therefore" or "because" with limited structural differences within in the dependency tree. The 50% correctness ratio for "därför att" in 2 were obtained from 200 validated specimens where no additional false negatives other the two mentioned meanings were obtained.

4.2 Robustness

Separation of training data and test data is a well known concept in terms of general research, however when manually elicitating the connectives no premeditated effort was undertaken for this kind of separation. This has probably negatively affected the results. The different rules extracted

for the different constituents seems to indicate this as well as different cue words has varying degrees of coverage and correctness. Because the actual meaning of words varies depending on context some of these contexts were probably not encountered before the rules were extrapolated. Cue words such as "eller" seems to be easily modelable as good coverage and low levels of false positives were obtained despite simplistic rules, whereas the cue word "inte" has high coverage and high degree of false positives, indicating an overly simplistic ruleset and too few samples before rule extraction.

5 Discussion

5.1 Model validation

The model seems to be valid for simplistic sentences where the sentences are constructed from proper nouns with a presumed lower probability of false positives than indicated by the results, but given the ruleset obtained the results are brittle as small lexical differences may make the text unparsable.

The rules implemented were sufficient to visualise simplistic examples but balked at infrequent occurrences that weren't encountered in sufficient quantities.

The method described in this project is sufficiently adequate for further study but would need to be less reliant on human intervention for rule creation as humans, as proved by some of the created rules, are insufficient rule makers.

5.2 Support automatic information extraction

When trying to extract information out of small quantity of text it might be hard to get the information if the fact is locked within an argumentation. As swedish is a small language with relatively low quantity of written digitized material, support in form of sentential logic extraction might help.

5.3 Combine with entity extraction for argument validation

In an application for validating the reasoning behind an argumentative text, for instance in an opinion piece, the use of sentential logic might be a crucial tool in analysing. The formal nature of sentential logic will generally aid in validating truthness in text, given that the text can be properly deducted and is logically sound. A common pitfall in data mining in text is entity ambiguity; such as when the parser is unable to properly relate a generic term such as "statesman" from an autobiography of a great politicial. We believe the dependency structure, given that the structured is used similarly as have been done in this report, will aid with the disambiguation due to the linked logical relationship between entities in the dependency tree. Another common error is equivocation; where multiple meanings are applied to an object, such as "No cat has two tails. A cat has one more tail than no cat. Therefore a cat has three tails"; this is a logical reasoning pitfall which is highly unlikely the method described in this project would notice.

6 What could have been done better

The general modus operandi of this project has been one of constant amazement; where each weeks progress has been hindered with the later realization that it's either already been tried or our approach would be limited by the available project time. The following section describes in no particular order musings realized near the end of the project of changes either to the process or possible avenues of modification given more time.

6.1 Training and test sets

When the project started the fact that text from swedish wikipedia was going to be used was already clear, thus it would has been wise to back from the beginning split the text into training and test sets. In fact the text set was split for convenience so the step to make the division would have been easy and the amount of text was well enough.

6.2 Methodology is manual, automate process

As the project carried on it become clear closer to the end that it would probably have been possible to automate some of the analysis. For instance the work of analysing in- and outgoing edges to split the cue words in different categories could have been done with some automated frequency analysation. Our approach were based on statistical analysis followed by model creation, whereas had we marked correct cue words in the corpus and thereafter used this as a golden standard general regression models could probably have been used. As it stands the validated data may not be viewed as a golden standard as too much human interference has been applied.

6.3 Narrowing down the scope of the project

When the project took off it aimed to solve a quite much bigger problem, partly in an area where we lacked proper domain knowledge. If the project would have been narrowed down earlier, the planning probably would have been smoother as we were unable to neither fully grasp the enormity of our situation nor our limited time span.

6.4 Pre Setup development environment

Quite some time were spent on solving issues regarding the development environment. An example is the combination of SVN (Subversion) and huge files, hours were spent on checking in and out of the huge corpuses. Much time and frustration would have been spared if more thought would have been given on this problem at an earlier stage.

References

Lennart Andersson. 2012. Diskreta strukturer. In *Föreläsningsanteckningar EDAF10*. Department of Computer Science at Lund University.

Sabine Buchholz and Erwin Marsi. 2006. Conll-x shared task on multilingual dependency parsing. In *Proceedings of the Tenth Conference on Computational Natural Language Learning*, pages 149–164. Association for Computational Linguistics.

Johan Hall, Joakim Nivre, and Jens Nilsson. 2007. A hybrid constituency-dependency parser for swedish. In *Proceedings of NODALIDA*, pages 284–287. Citeseer.

Robert Östling. 2012. Stagger: A modern pos tagger for swedish. In *The Fourth Swedish Language Technology Conference*.