Extracting Knowledge from Wordnet: Artifacts and Their Roles

Ishan Agarwal Advisor: Prof James Pustejovsky

Brandeis University, Summer 2017

Background

The problem we try to address is to classify words for concepts in language that have a particular pre-decided role, in some sense things that were made for some particular purpose and have a so- called 'telic role'. For example words like 'air' or water lack a specific telic role, but words like 'hammer' have one. In generative lexicon theory such words are called artifacts and are distinguished from other kinds of nouns.

Objectives

We aim to extract from the word net a list of words that have such a telic role as well as to extract the role in question from a human generated definition of the word. In particular we seek a noun and verb that describe the role of the concept represented by the artifact. We test and score our algorithms against a small test set where the telic role has been extracted from the definition by hand.

Methods

We first extracted the artifacts from wordnet by simple searching for keywords such as 'used' or 'designed' We used a few different algorithms to attempt to extract the telic role:

- 1)We simply return the noun verb pair closest to a keyword in the gloss which is after the said keyword. This is based on the heuristic that roles are generally consist of a noun and a verb and if the sentence construction is simple they are likely to be in proximity to a word signalling the presence of a telic role and after it.
- 2)We use a parse tree and simply proceed down the first path from a keyword uptill a certain constant. This modifies the previous idea of spatial proximity to that of proximity in the chain of dependencies.
- 3)We use a dependency parser to generate parse trees for each gloss. Manually we look at many glosses and find typical paths from a keyword to telic roles. We list such paths and our algorithm searches down such paths and returns words along the path that is most consistent with some path in our list.
- 4)Finally we try a combination of the above techniques in which we use the parse tree but proceed along it till a noun -verb pair is reached to avoid not looking down the correct path in case there are multiple paths or the other problem of not extracting words at the right depth of the path.

Results

We use both a strict and relaxed scoring for a sample size of 75 word-gloss pairs. The combined recall and precision score is the harmonic mean.Note that out of 75 extracted words only some were not really artifacts leading to a sample size of 63 actual artifacts. The difference is counted as true negatives in the scoring. The relaxed scheme graded the answer as correct if the output contained either a noun or a verb from the correct telic phrase. The strict scheme awarded 0.4 score for only a matching noun or only a matching verb but 1 if both matched. If there were extraneous nouns or verbs , we counted that as incorrect noun or verb part.

For relaxed grading:

algorithm	precision	recall	combined score
1	82.3	68.3	74.6
2	83.4	69.6	75.9
3	89.2	84.8	86.9
4	89.2	84.8	86.9

For strict grading:

algorithm	precision	recall	combined score
1	72.7	61.9	66.8
2	73.5	60.8	66.5
3	77.6	68.5	72.8
4	80.2	74.1	77.0

Further discussion

- Note that the recall is universally found to be worse than the precision. This is a natural consequence of the inability of our algorithms to correct themselves if they find something with a keyword but which is not really an artifact. I believe that this can be solved by a small amount of pattern matching/case checking.
- Another feature to note is that the performance of all our algorithms gets significantly worse under the strict grading. In some sense this is because, as was the initial heuristic part of the correct telic role is in proximity, either physically or in the dependency parse tree to the marker word we are using for identifying the word as an artifact. Basically since the relaxed scheme grades a role as correct even if it only partially finds the role, it naturally results in higher scores.
- Note that under relaxed grading the combined algorithm does no better than algorithm 3. This was expected as it basically refines searching along the parse tree and the relaxed grading does not differentiate between getting a part and the whole of the telic role.

Ideas for future directions

As already discussed, we can seek interpretations of the results. Furthermore this is only a very small step in building an ontology/knowledge base compatible with the principles of GL. Several tools exist (see CoreLex) and more need to be developed, especially for other classes of words (eg verbs and adjectives). In particular it is possible that the techniques we use or the data extracted may be useful in distinguishing artifacts from natural objects that also have a specific primaypurpose but were not designed for this thing in particular. Apart from this there is also the possibility that an endeavour to create better tools such as this will lead to theoretical insights; attempting good and natural classification/role extraction algorithms for words may lead to insight into the underlying theory of lexicons and semantics which may help deal with problems of ambiguity ,such as polysemy, in natural language processing.

References

I studied material from the following to familiarize myself with the Generative Lexicon theory and some basic ideas in computational linguistics:

- The Generative Lexicon ; James Pustejovsky
- The Language of Word Meaning; edited by Pierrette Bouillon and Federica Busa
- Speech and Language Processing ; Daniel Jurafsky and James H. Martin

The following was used as a reference for WordNet:

- WordNet An Electronic Lexical Database ; Christiane Fellbaum

Acknowledgements

I would like to thank the following people for their support over the last two months without which none of this work would have been possible:

- Professor James Pustejovsky who introduced developed the GL theory and introduced me both to it and to this particular problem
- Marc Verhagen who guided me through several technical difficulties and suggested several of the methods used
- Nikhil Krishnaswamy who for his help and advice about innumerable matters both technical and practical
- Wanda Weinberger, Jessie McShane, Steven Karel and all the people at Brandeis involved with the Brandeis IISc exchange programme for their hospitality and for making this research work ,and indeed this presentation, possible.

Further material

Related material including some of the python code, raw output txt files, as well as more details in the form of full project reports, can be found at my github: https://github.com/IshanAgarwal/Computational-Linguistics-Brandeis-Summer-2017 (please look at the Readme for specifications regarding versions of software used)