THESIS WORKING TITLE

Dorus Keijzer

September 22, 2023

1 Introduction

1.1 Motivation

In the model developed by Groschwitz et al. (2018) this lexicalization step is performed by a Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997) model. In the field of natural language processing, LSTMs have been supplanted by more recent models (discussed in the literature review below), which gain a more fine grained insight into language by being trained on large corpora of text before being fine-tuned to fit a specific task. Replacing the original lexicalization model(which performed below expectation on this task) by a fine-tuned modern language model could elevate the performance of the entire parser.

1.2 Academic relevance

This thesis is located within the discipline of natural language processing, a sub-field of artificial intelligence. AMR parsing is a process of interfacing between natural language and a semantic representation that is readable to both people and computers. Such meaning representations could for instance allow us to create knowledge banks based on natural language input, that could be used for to perform inference on.

2 Background

2.1 AMR format

Abstract meaning representation (Banarescu et al., 2013) is a formalism that attempts to capture the semantics of a natural language sentence in a graph based format. AMRs are rooted, directed, node-labeled and edge-labeled graphs. The nodes of these graphs represent the core concepts in the sentence, the edges convey how these concepts are related. The labels of nodes representing nouns are variables, often named after the object they represent, in the example below, the node representing ”orca” is represented as ”orca”. Predicates are represented using Ontonotes (Hovy et al., 2006), which disambiguates predicates by asigning every sense of a word a different number: grill-01 refers to the act of cooking on an open fire, grill-02 refers to an interrogation. Names, dates and numbers have their own respective formats. AMRs can also be represented as a conjunction of logical propositions

2.2 AM-parser

The AM-parser (Groschwitz et al., 2018) is a parser that converts a natural language sentence to its corresponding AMR graph, using the AM-algebra developed by Groschwitz (2019).

2.2.1 AM-algebra

The AM-algebra is an algebra The smallest possible term in the AM algebra are annotated s-graphs (or: AS-graphs), which convey the meaning of *one* core concept through giving what the sense of the word is, and to which *sources*, i.e. types of words it could be related to in which way. These AS-graphs can be combined in two different ways: *application* and *modification*. Whether two graphs can combine is determined by a type system. Application fills in the argument of a given predicate, modification adds a modifier to a given graph. In both operations, if the two terms that are being combined share a source s, the nodes marked with s are unified with eachother.

2.2.2 Supertagger

2.2.3 Architecture

Because this problem is NP-complete, ... there are two parsers ... same results ... this thesis uses the following parser, based on the fact that this one has been shown to work. ???

2.2.4 lexicalization

2.3 Long short-term memory

2.4 pre-trained sequence-to-sequence language models

The transformer architecture (Vaswani et al., 2017) made it possible for language models to be trained in parallel, rather than token-for-token as in LSTMs. This, combined with the ease of scalability of these models allowed for transformer-based models to be trained on much larger data-sets compared to previous methods. This inspired researchers to apply transfer-learning techniques that had previously been successful in the field of computer vision (Yosinski et al., 2014) to language models. Instead of training a new model for every task, researchers trained *one* model on vast amounts of data. This model thereby gained insight into the workings of language, which can be leveraged by further training this model on a specific task (fine-tuning). Because the pre-trained model retains its understanding of language from the pre-training step, the fine-tuning can be performed on a much smaller database, in less time and with better results compared to training a model specially for the task at hand [citation needed]. Examples of these kind of models are BART, BERT and GPT. (Devlin et al., 2019; Lewis et al., 2019; Radford et al., 2019). For this thesis,

2.4.1 BART

BART (**BART**) is a sequence-to-sequence model based on a bidirectional encoder trained on the task of reconstructing a corrupted text. The text corruptions BART is trained on consist of the masking of random singular tokens, the deletion of random tokens, the random masking of multiple contiguous tokens, the shuffling around of random tokens within a sentence and rotating the starting token of a text.

2.4.2 T5

3 Methodology

3.1 Model

3.2 fine-tuning

3.3 Evaluation

The model will be evaluated using the commonly used the Smatch evaluation metric (Cai & Knight, 2013), as well as the more fine grained GrAPES evaluation suite (**GrAPES**). These results will be compared to the results obtained by **GrAPES<empty citation>**.

3.4 Smatch

Smatch measures the degree of overlap between two AMRs. Because the names of AMR variables can be chosen freely , it is not straightforward to compare variables on their given names. When presented with these AMRs in logical form:

$$AAAAAAAAAAAAAAAAAAAA$$

it is not clear if parser 1 intended for x to refer to what parser two calls "John", and y to "Sue". Therefore, Smatch compares possible assignments and returns the F-score of the best possible overlap it is able to find. Here, Smatch will be used to evaluate the AMRs made by the parser against annotated gold-standard test data. [WELKE DATA]

3.5 GrAPES

GrAPES is a more fine grained evaluation suite that has been devised to allow for focus on important linguistic features that Smatch cannot adequately find. Because

4 Results

4.1 SMATCH

4.2 GrAPES

5 Discussion

5.1 Further research

As the original model already creates words embeddings through a modern language model (namely BERT),

References

Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., & Schneider, N. (2013). Abstract Meaning Representation for sembanking. *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186. https://aclanthology.org/W13-2322

Cai, S., & Knight, K. (2013). Smatch: An evaluation metric for semantic feature structures. *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 748–752. https://aclanthology.org/P13-2131

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Groschwitz, J. (2019, January). *Methods for taking semantic graphs apart and putting them back together again* [Doctoral dissertation].

Groschwitz, J., Lindemann, M., Fowlie, M., Johnson, M., & Koller, A. (2018). AMR dependency parsing with a typed semantic algebra. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1831–1841. https://doi.org/10.18653/v1/P18-1170

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*, 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). OntoNotes: The 90% solution. *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, 57–60. https://aclanthology.org/N06-2015

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2019). BART: denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *CoRR*, *abs/1910.13461*. http://arxiv.org/abs/1910.13461

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. https://api.semanticscholar.org/CorpusID:160025533

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. http://arxiv.org/abs/1706.03762

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *CoRR*, *abs/1411.1792*. http://arxiv.org/abs/1411.1792