Thought on a new word embeddings model

Jianyu Zhan
nasa4836@gmail.com

1 Motive

Doing word embeddings is a prerequsite and important procedure in all NLP tasks. The mainstream word embeddings schemes like Word2vec[2] uses prediction based method from corpus to derive semantic word embeddings. However, this scheme uses one embedding vector for a word, which doesn't take into account context info and mononymy and polesymy problem[3, 1]. What's more, in all current language models, word embeddings seems not to play a central role, but merely a preprocssing step in NLP tasks that just transforms human readable text into digital vectors. Instead we put a lot of efforts into devising sophiscated models for specific NLP tasks.

I think word embeddings should have played a pivotal role in NLP for two reasons. One is it ussually spans huge vector space dimensions, which have really high information entropy that could have been used to encode more information about the text context of words. The other is that word embeddings is a natual medium that could convey more information to share among various NLP tasks.

So my thought is that ideal word embeddings should carry both syntactic and semantic information. We now have the elegant Word2vec model, that could carry compositable semantics of words, but we don't have a more versatile scheme that also encodes the syntactic information, which is critical in NLP because we more often need to disambiguate words meaning by the context they are in than just using a word's standalone meaning.

I have did preliminary research and found that one work that is close to my goal is Sense2vec[4]. But the problem is that it still disregards the pivotal role of word embeddings, and just leverages part-of-speech inforamtion to generate word embeddings with context infomation built in inexplicitly. So it is still hard to extract this syntactic information from the learned word embeddings.

With the important role of word embeddings in mind, I plan to explore a direction of creating word embeddings with explicit syntactic information encoded, alongside semantic information. While we have this, it is likely quite a bunch of NLP tasks could benefit from it.

2 Proposed scheme

The rough idea is to divide the huge word embeddings vector space into *direct product* of subspaces. which are orthogonal to each other. A subspace of it could be used to encode the semantic information, like what Word2vec does, the rest of subspaces are used to encode the syntatic information - the *part of speech*. Figure 1 is a schematic model.

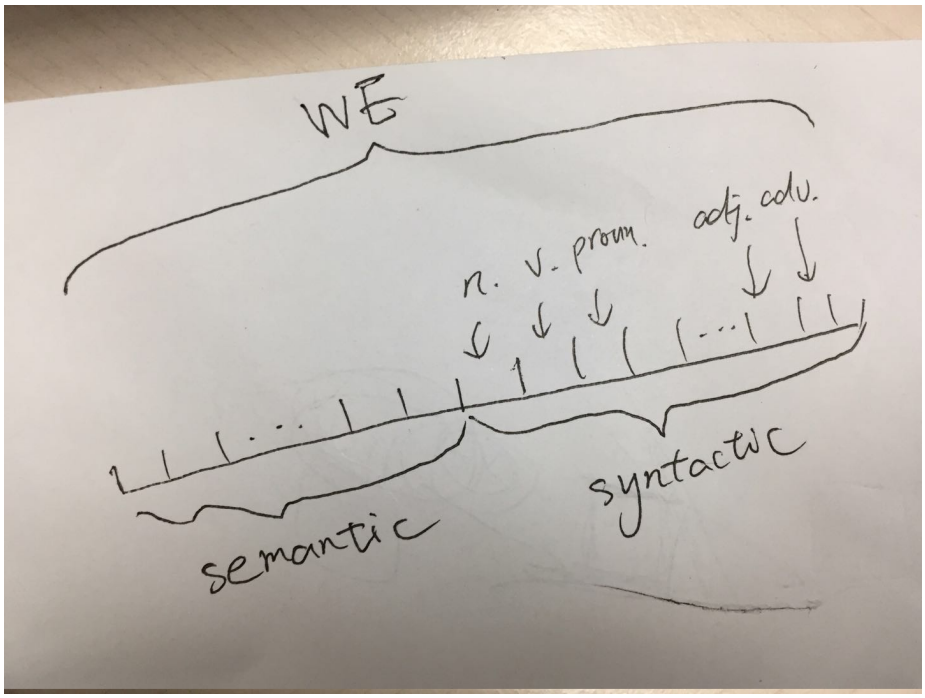

Figure 1: Schematic division of subspaces of word embeddings

Using orthogonal subspaces has such advantages. One is that the word embeddings could be learned seperately or parallel, for example we can learn the semantic embeddings using Skip-gram[2] and learn the syntactic embeddings using some PoS model seperatelly. The other one is that given a word embedding of a word, we can easily get its semantic part by projecting it into the semantic subspace, the same for extracting specific PoS information.

3 Preliminary thought on impletation details

A naive method is to leverage the existing Word2vec model and PoS model for learning the respective embeddings seperately. I am reading PoS papers in order to design a more detailed scheme.

To induce the desired subspace division, we can turn the learning objective function into a contrained optimization problem by preferring vectors that with specific part masked out.

4 Problems that need more thinking

Usually a word has more than one meaning from the perspective of PoS, or more than one *sense* using Sense2vec terms. So it is better to have different vector for different PoS for a word. In this new scheme, we natually have this, since we explicitly encode the PoS in the syntactic subspace. One problem is that usually when a word change its PoS category, its morph changes. For example in inflectional language like English, we change a word's suffix to change its PoS category. So probably we should induce the semantic part to learn the subword or the stem encoding, and for the syntactic part to learn the affix encoding.

The other problem is that how to divide the syntactic subspaces for different PoS categories. Because PoS categories are not equally important, so they should be arranged structually. For a specific sentence, its backbone struct is *subject + verb + object(SVO)*, in which *subject* and *object* could recursively has similar struct. For an example, we have *SVSVO*, and we substitue it with PoS categories, then it could be *noun + verb + pronoun + verb + noun* or *pronoun + verb + pronoun + verb + noun*, etc. We can see different PoS categories have a structual hierarchy. So my plan is to divide the syntactic subspaces to reflect this structure. And if we have one, we probably could have a more concise sentense encoding by organically "summing up" its constitutent word embeddings, instead of naviely concatenating all words' embeddings. And this might help simplify some model of NLP tasks, for example, we might get rid of the encoder-decoder model in NMT.

References

[1] Huang, E. H., Socher, R., Manning, C. D., and Ng, A. Y. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1* (2012), Association for Computational Linguistics, pp. 873–882.

[2] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

[3] Reisinger, J., and Mooney, R. J. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010), Association for Computational Linguistics, pp. 109–117.

[4] Trask, A., Michalak, P., and Liu, J. sense2vec-a fast and accurate method for word sense disambiguation in neural word embeddings. *arXiv preprint arXiv:1511.06388* (2015).