Antonym Detection (Papers)

Li, J., Hu, R., Liu, X. et al. Neural Comput & Applic (2019). https://doi.org/10.1007/s00521-019-04071-6

- Basic word vector models (such as word2vec) have utilized the word-level co-occurrence information either implicitly or explicitly; but they did not take some fine-gained between-word relation.
- To improve the word representations, we can introduce external resources into models during the learning or post-processing step to specialize word embeddings. But, context words represent the syntagmatic relations, while synonyms, antonyms and hypernyms represent paradigmatic relations.
- Novel models attempt the task by maximizing the similarity between synonyms and minimize the similarity between antonyms. However, from the general standpoint, two antonymous words still belong to the same category (same frame) and are highly relevant. Minimizing similarities of antonyms might result in uncontrollable vector movement and, thus, negatively affect the global semantic distribution.
- Proposed models - Introduces paradigmatic relation into Skip-gram model and shortens the distance between target word and its synonyms by controlling their movements in both unidirectional and bidirectional. By enabling the synonyms to move closer to each other, the distance between antonyms will also become more noticeable. PPDB, a thesaurus is used to offer distant supervision to the Skip-gram model.
- $L(H) = Pr(w_1, .., w_n|w_t) + \alpha . J(w_t, w_{syn})$ where α ranges from 0.1 to 0.2 and Pr (from Skip-gram) represents the predictive probability of context words conditioned on the target word w_t.
- **UMT** - randomly selects a synonym of the target word and moves the target word toward the synonym by updating the target word vectors by $L(H)$.
- **UMS** - all the corresponding synonyms of a target word are moved together toward the target.
- **BMTS** - randomly chooses one synonym to calculate the loss function $L(H)$. Then move the target word vector as well as the synonym word vector.

Dou Z., Wei W., Wan X. (2018) Improving Word Embeddings for Antonym Detection Using Thesauri and SentiWordNet. In: Zhang M., Ng V., Zhao D., Li S., Zan H. (eds) Natural Language Processing and Chinese Computing. NLPCC 2018. Lecture Notes in Computer Science, vol 11109. Springer, Cham

- A pair of antonyms, for example, “long” and “short”, tend to appear in similar context environment. This leads to the serious problem that it is extremely difficult to discriminate antonyms from synonyms using word embeddings.
- key characteristic of current word embeddings: vectors of related words are supposed to have close directions while unrelated words correspond to vectors in opposite directions. The cosine distance between completely unrelated words is close to −1, while the cosine distance between a pair of related words, including both synonyms and antonyms, is close to 1.

- SentiWordNet - a dataset which labels each English word with a corresponding 3-dimensional vector. The three components of the vector respectively represent the positive emotion rate, negative emotion rate and objective rate of the word with their sum 1. Pos(w)+ Neg(w)+ Obj(w)=1.
- Proposed model - Make the cosine distance of synonyms close to 1 and the cosine distance of antonyms close to −1, whereas the cosine distance between unrelated words is close to 0. Then a pair of words with a negative cosine distance close to −1 are likely to be antonyms.
    - **Skip-Gram Model with Negative Sampling with modified objective function**. The objective funtion is by squaring the functions inside the sigmoid function.
    - **Word Embedding using Thesauri Dataset Information with Max Margin Framework**. Max Margin Framework which can significantly decrease the risk of overfitting. Maximization of objective function makes the similarity score between synonyms very high and that between antonyms very low.
    - **Word Embeddings Based on SentiWordNet**. Assumption is that those senti-synonyms should have not only high senti-similarity score but also high word similarity score and vice versa.
        - Semi-similarity - normalized dot product of [Pos(w1),Neg(w1)] and [Pos(w2),Neg(w2)].
- Datasets
    - **GRE antonym questions dataset:** May not be applicable to FN task because there's no cutoff necessary as the task is to choose the candidate who has the lowest similarity with the target word
    - **WordSim353**: Experiments are conducted on STS Benchmark where performance is measured by the Pearson correlation of machine scores with human judgments.

Asif Ali, Muhammad & Sun, Yifang & Zhou, Xiaoling & Wang, Wei & Zhao, Xiang. (2019). Antonym-Synonym Classification Based on New Sub-space Embeddings.

- embedding based approaches (i.e. neural networks, matrix factorization, co-reference chains + skip-gram, distributional lexical contrast information + skip-gram, thesaurus + distributional information, skip-gram + WordNet)
    - Limitation: incapable of discriminating between different lexico-semantic relations
- Pattern-based approaches - capturing lexico-syntactic patterns from large scale text corpora (lemma, POS-tag and dependency label)
    - huge overlap in feature space (i.e., lexico-syntactic patterns) such as noun-compounds across different classes, which hinders improvement in performance.
    - Pattern based methods yield a low recall owing to the lexical variations and the sparsity of lexico-syntactic patterns, which limits the information sharing among semantically similar patterns.
    - Have to use a large-scale text corpus to combat sparsity, which dras- tically increases the computational overhead.
- Proposed model:

    - **Distiller**: two different neural-network encoders (enc_s, enc_a) to project pre-trained embeddings to two new sub-spaces (synonym SYN and antonym ANT sub-spaces) in a non-linear fashion. This also models the symmetry and trans-transitivity preserving antonym relation using both ANT and SYN.
        - *Loss function for Synonyms*: models the symmetry and transitivity preserving synonym relation in SYN by using the following loss function $L_S = \sum_{(a,b)} ReLU(1 - f(a,b)) + \sum_{(a',b')} ReLU(1 + f(a',b'))$ where (a', b') is the pair of irrelevant words and $f(a,b) = tanh(< enc_s(a), enc_s(b) >)$. It is a margin based loss function that makes synonyms close to each other and irrelevant pairs away from each other.
        - *Loss function for Antonyms*: models the symmetry and trans-transitivity for antonym pairs by using the loss function $L_A = \sum_{(a,b)} ReLU(1 - g(a,b)) + \sum_{(a',b')} ReLU(1 + g(a',b'))$ where (a', b') is the pair of irrelevant words and $g(a,b) = tanh(< enc_a(a), enc_s(b) >)$.
        - *Loss function for the entire model* = $L_S + L_A + L_M$ where $L_M = -(1/N) \sum_{i=1}^{N} log(\hat{p}(y_i|a_i, b_i))$ (an entropy-loss function).
            - $\hat{p}(y_i|a_i, b_i) = softmax(Wx + b)$ where $x = [x_{syn\_score}, x_{ant\_score}]$
            - $x_{syn\_score} = cos(enc_s(a), enc_s(b))$
            - $x_{ant\_score} = max(cos(enc_a(a), enc_s(b)), cos(enc_a(b), enc_s(a)))$
    - **XGBoost classifier** - input features: x_{syn_score}, x_{ant_score}, $cos(a,b)$, $has\_negation\_prefix(a,b)$
        - $x_{syn\_score} = cos(enc_s(a), enc_s(b))$
        - $x_{ant\_score} = max(cos(enc_a(a), enc_s(b)), cos(enc_a(b), enc_s(a)))$
        - $cos(a,b)$ = cosine similarity of the pair of pre-trained word embeddings
        - $has\_negation\_prefix(a,b)$ = 1 if one of the word contains the prefix from the following set of negation prefixes: {de, a, un, non, in, ir, anti, il, dis, counter, im, an, sub, ab}
- Dataset:
    - https://github.com/nguyenkh/AntSynDistinction
- Baseline models:
    - **Discourse Markers** [Roth and Schulte im Walde (2014)] - use indicators of relations and lexicon-syntactic patterns such as dependency and POS to design vector space
    - **Symmetric Patterns** [Schwartz, Reichart, and Rappoport (2015)] - extract symmetric patterns of sentences from plain text
    - **AntSynNET** [Nguyen, Schulte im Walde, and Vu (2017)]
- Characteristics of Proposed Model:
    - Language-agnostic
    - limited by multiple senses (polysemous words) and rare tokens
    - increase in word frequency decreases model's performance because high frequency words are polysemous.
    - allows appropriate re-ordering of the sub-spaces to capture relation-specific

characteristics.
    - Distiller is a flexible and efficient choice for lexical-semantic analysis requiring no pre-processing and little linguistic knowledge due to the wide availability of pre-trained embeddings

---

Questions

- How to evaluate the performance of different antonyms detection models other than manual checking?
- What does it mean by "overlapping in feature space"? Quote: "A major limitation of the pattern based methods is the huge overlap in feature space (i.e., lexico-syntactic patterns) across different classes, which hinders improvement in performance. For example, a commonly confused pattern is the noun-compound, which can be used to represent both the synonym pairs (government administration; card board) and the antonym pairs (graduate student; client server)."
- Should I manually create the dataset of antonymous pairs of framenet?
- One topic that is worth pursuing using FrameNet: polysemous words