B06102020 楊晴雯

2021 Fall Computational Linguistics

Prof. Shukai Hsieh

January, 14 2021

Chinese Wiki-based Word Sense Disambiguation

I. Introduction

The issue of polysemy has several solutions in the field of computational linguistics or NLP, and most of them profit from already-existing word-sense databases. However, given that a sense database with widely-agreed, up-to-standard annotation (eg. English WordNet) is very rare, Sara Tonelli et al. (2013) mapped Wikipedia disambiguation senses to the semantic frames in FrameNet in the wish of automatic frame annotation. Their method gives me an idea that a Chinese word-sense-disambiguation (WSD) tool could be developed based on the rich manually-annotated disambiguation pages from Wikipedia. In fact, several related works (Dandala et al., 2013) have been presented proving Wikipedia to be a reliable source for WSD on English SENSEVAL-2 and SENSEVAL-3 datasets. In this paper, I investigate a Chinese Wiki-based WSD system and build sense classifiers with a statistic-based and a neural-based algorithm respectively, and finally their performances are compared.

II. Dataset

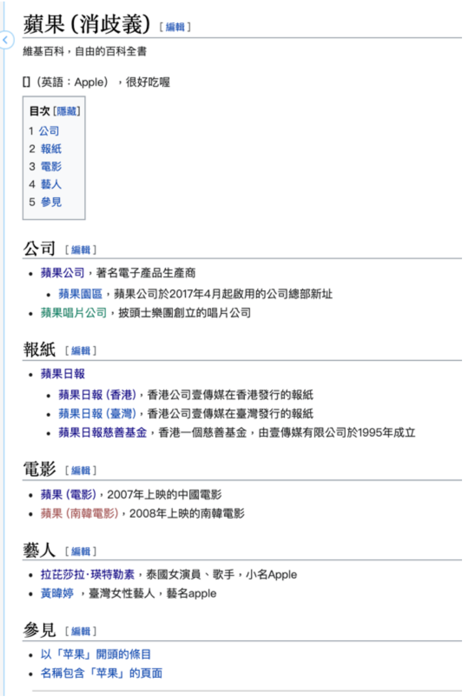
The dataset collection and processing workflow is as follows. Firstly, a phrase list is selected and the words within are checked with the following conditions: (1) if the word has a corresponding Wikipedia page, (2) if its Wikipedia page has an out-link to its disambiguation page, and the disambiguation page has more than one sense entry listed. The phrase list[1] picked has 55,729 words. It first undergoes a conversion from simplified Chinese to traditional Chinese, and then the above elimination conditions eliminate about 95% of words, leaving only 2,364 words.

After reviewing some of the data collected, it is noted that Wiki disambiguation page is careful and critical in including and defining entries—see Fig1 for example, “蘋果日報” has 4 different entries dedicated to the news agency of different locations

[1] Joseph, C. (2014). 現代漢語常用詞表. https://gist.github.com/indiejoseph/eae09c673460aa0b56db.

and its affiliations (“蘋果日報”, “蘋果日報（香港）”, “蘋果日報（臺灣）”, “蘋果日報慈善基金”). This arrangement is very helpful for readers to be quickly directed to the terms of their interest with just a click from the disambiguation page. For WSD, however, there’s usually no need to differentiate between the meanings of these fine-grained senses; to be able to differentiate between the sections (“水果”, “公司”, “報紙”, “電影”, “藝人”) is enough. In other words, the section (denoted by coarse-grained senses) is a more proper granularity level to choose than the sense entries (denoted by fine-grained senses) for the WSD task.

Fig.1. Wiki disambiguation page for “蘋果.”



蘋果 (消歧義) [編輯]

維基百科，自由的百科全書

[] (英語：Apple) ，很好吃喔

目次 [隱藏]
1 公司
2 報紙
3 電影
4 藝人
5 參見

公司 [編輯]

- 蘋果公司，著名電子產品生產商
    - 蘋果園區，蘋果公司於2017年4月起啟用的公司總部新址
- 蘋果唱片公司，披頭士樂團創立的唱片公司

報紙 [編輯]

- 蘋果日報
    - 蘋果日報 (香港)，香港公司壹傳媒在香港發行的報紙
    - 蘋果日報 (臺灣)，香港公司壹傳媒在臺灣發行的報紙
    - 蘋果日報慈善基金，香港一個慈善基金，由壹傳媒有限公司於1995年成立

電影 [編輯]

- 蘋果 (電影)，2007年上映的中國電影
- 蘋果 (南韓電影)，2008年上映的南韓電影

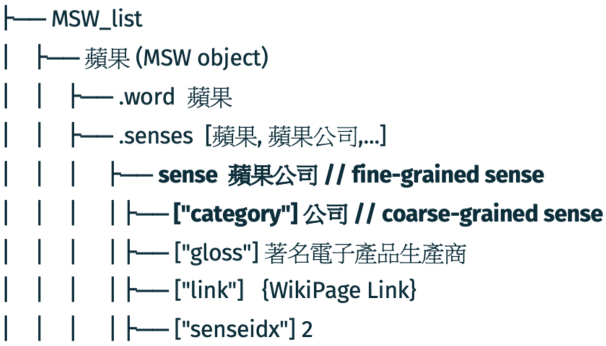
藝人 [編輯]

- 拉茫莎拉·瑛特勒素，泰國女演員、歌手，小名Apple
- 黃暐婷 ，臺灣女性藝人，藝名apple

參見 [編輯]

- 以「苹果」開頭的條目
- 名稱包含「苹果」的頁面

Fig.2. Dataset structure.



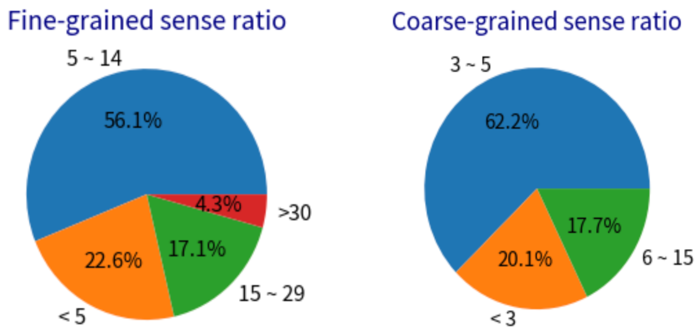
```
├── MSW_list
|   ├── 蘋果 (MSW object)
|   |   ├── .word  蘋果
|   |   ├── .senses  [蘋果, 蘋果公司,...]
|   |   |   ├── sense  蘋果公司 // fine-grained sense
|   |   |   | ├── ["category"] 公司 // coarse-grained sense
|   |   |   | ├── ["gloss"] 著名電子產品生產商
|   |   |   | ├── ["link"]  {WikiPage Link}
|   |   |   | ├── ["senseidx"] 2
```

Since the disambiguation pages across the Wiki corpus do not always have section structures, only the 164 words (0.3%) having sections are kept in the final dataset. The dataset is denoted by a MSW (multi-sense word) dataset, and the words within are denoted by MSWs (multi-sense word).

The final dataset is organized as Fig. 2. Every MSW has a property $.senses$, which is a list keeping all its fine-grained senses; each sense is a dictionary with keys “category”, “gloss”, “link” and “senseidx.” *Category* is the coarse-grained sense that it belongs to; *gloss* is the one-sentence description extracted from the disambiguation page (if the description is absent, the first 50 words from summary in the sense page is used instead); *link* is the Wikipedia page link and *senseidx* is the sense index arbitrarily assigned.

In average, one MSW has 11.03 fine-grained senses and 3.95 coarse-grained senses. The MSWs having 5~14 fine-grained senses account for the highest ratio (56.1%), while the MSWs having 3~5 coarse-grained senses account for the highest ratio (62.2%) (see Fig.3.).

Fig.3. Sense ratio pie charts.



III. Methodology

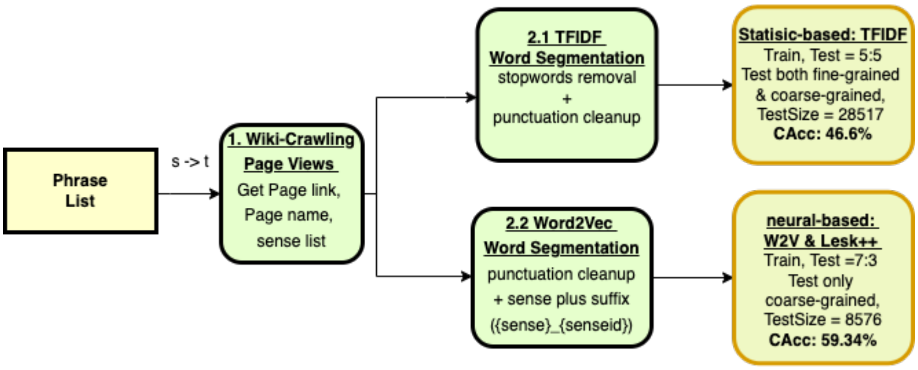
The pipeline of the experiment is shown in figure 4. The first steps of phrase-list picking and Wiki-crawling are done in dataset collection stage. By now each MSW has multiple senses, and each sense comes with a sense-file (i.e. the corresponding Wikipedia page content). With the assumption that the target MSW w_i appearing in s_j sense-file is fully disambiguated into that sense (s_j), namely:

$$\forall I \in doc(w_i, s_j) \text{ } and \text{ } w_i \in I,$$
$$gold\_label(w_i) = s_j$$

, we obtain a corpus with sense annotation for the target word w_i. The pipeline at WSD stage then diverges in 2 ways: Statistics-based TFIDF and neural-based: word2vec & Lesk++[2] (Oele & Noord, 2018). After segmentation, distinct preprocessing methods are carried out respectively. For TFIDF, a Chinese stopword list[3] is used for stopword removal, and punctuations as well as URLs are removed. For word2vec & Lesk++ (hereafter denoted by W&L), the sense-files undergo the punctuation and URL cleanup, and are sent to embedding training. Note that the procedure of adding suffixes to different senses in one word (in the form $\{targetMSW\}\_\{senseidx\}$) is done to ensure that sense embeddings are trained separately.

[3] https://github.com/stopwords-iso/stopwords-zh/blob/master/stopwords-zh.txt

Fig.4. Experiment pipeline.



(I) Statistic-based: TFIDF

The first algorithm is the all-time classic TFIDF method. It uses term frequency of w_i in document d_j and (inverted) document frequency to give common words high weight, rare words low weight; while simultaneously restrains the weights of those common words across all documents (eg. stopwords, determiners in English). In this way, TFIDF determines a score for how important a term is to a document. Mathematically, tf (term frequency) is

$$tf_{i,j} = \frac{n_{i,j}}{\sum_{k=1} n_{k,j}}$$

where $n_{i,j}$ represents the count of word w_i in document j; idf (inverted document frequency) is

$$idf_i = \log\left(\frac{|\mathrm{D}|+1}{|j{:}t_i \in d_j|+1}\right)$$

where $|\mathrm{D}|$ is the number of documents and $|j{:}t_i \in d_j|$ is the number of documents where word w_i appears. The final score is the product of two.

$$tfidf_{i,j} = tf_{i,j} * idf_i$$

Given an input sentence I containing the MSW w, among all its senses, this algorithm chooses s_j to be the prediction, such that

$$score(d_j) = \sum_{t \in I, t \neq w} tfidf(t_i, d_j) \quad is\ maximized$$

where d_j is the sense-file $doc(w_i, s_j)$. Note that the score for the MSW to be disambiguated is not included.

(II) Neural-based: Word2vec & Lesk++ (W&L)

Oele and Noord (2018) proposed an improved version of the classic WSD Lesk algorithm, called Lesk++. The traditional Lesk is based on statistics: for MSW w_i, Lesk counts the number of overlapped words in input sentence *I* and dictionary gloss for all s_j, and then predicts sense s_j with the highest count. The improved Lesk++ used in their paper as well as in the following experiment makes use of "a combination of sense embeddings, context embeddings, and gloss embeddings." Given input sentence *I* containing MSW w_i, *I* is transformed into a context vector C_w, and for each of w_i's potential sense s_j, the cosine similarity scores between the gloss embedding G_{sj} and C_w and between the lexeme embedding L_{sj} and C_w are computed and added together:

$$score(s_j, w_i) = \cos(G_{sj}, C_w) + \cos(L_{sj}, C_w)$$

While Oele and Noord's method uses AutoExtend embeddings trained on WordNet corpus and gloss-expansion techniques, our implementation is simpler: the word2vec suffixed sense embeddings serve as L_{sj}; G_{sj} is the average of word2vec embeddings of all $g \in$ gloss of s_j; C_w is the average of word2vec embeddings of $c \in$ input *I*. Note that this training scheme produces an un-suffixed embedding for MSW w_i, if w_i appears other than in its own sense-files as an un-disambiguated word, and C_w includes it when averaging the token embeddings.

(III) Parameter Setting

TFIDF has Train: Test = 1:1, that is, half of the sentences are used as sense profiles, while the other half are used for testing (28,517). W&L has Train: Test = 7:3; the sentences without target MSW removed from the corresponding sense files, which lowers the testing sentences count to 8,576. As for word2vec training, the embeddings are trained with embedding dimension = 200, context window = 7, minimum count = 3, epochs = 30. The final vocab size is 72,275.

IV. Result

Table1. Result for TFIDF and W&L.

	Coarse-grained counts				
	< 3	3~5	6~15	Overall	Test Size
TFIDF	54.87%	50.83%	40.69%	46.65%	28517
W&L	**63.53%**	**63.15%**	**48.46%**	**59.34%**	8576

The results shown in Table 1. suggest that word2vec & Lesk++ significantly outperforms TFIDF in all coarse-grained count intervals, and the overall accuracy is higher by 12.69%. This experiment result is not surprising, since TFIDF is based on statistics, it is quite easily deviated by unexpected stopwords not contained in the removal stage. Furthermore, it lacks the ability to distinguish in between the 2 coarse-grained senses "公司" and "報紙," as can be seen in Fig.5., where it mislabels 53.27% of the "公司" test sentences as "報紙." In comparison, W&L could distinguish very well between the two senses, showing better WSD ability with neural model's generalizability. Although some doubts exist in W&L in terms of the 8 and 4 sentences mislabeled to "main;" they do not appear to have WSD ambiguities according to my manual review. A testing procedure with more delicate recording scheme would have to be carried out to find out the reasons, but time prohibits.

The general correlational tendency is that the less coarse-grained senses a MSW has, the higher accuracy it obtains. Both TFIDF and W&L results show a strictly decreasing accuracies as coarse-grained counts grow, with a significant drop by over 10% from count interval 3~5 to 6~15. This phenomenon could be understood from simple statistics and data distribution skewness. See Fig.6. for the confusion matrices of "歌手, " in which both methods obtain high accuracy, but the recall for coarse-grained sense "綜藝節目" is very low in TFIDF, and the data is highly uneven in label distribution in both test corpora—guessing all sentences to be "綜藝節目" could make a strong baseline in this case.

Fig.5. Confusion matrix of “蘋果”: TFIDF (left) vs. W&L (right)

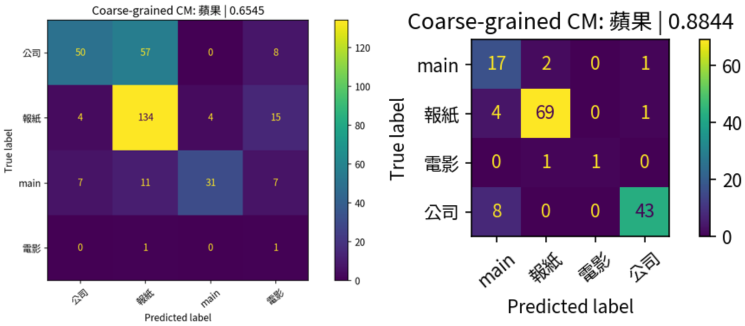
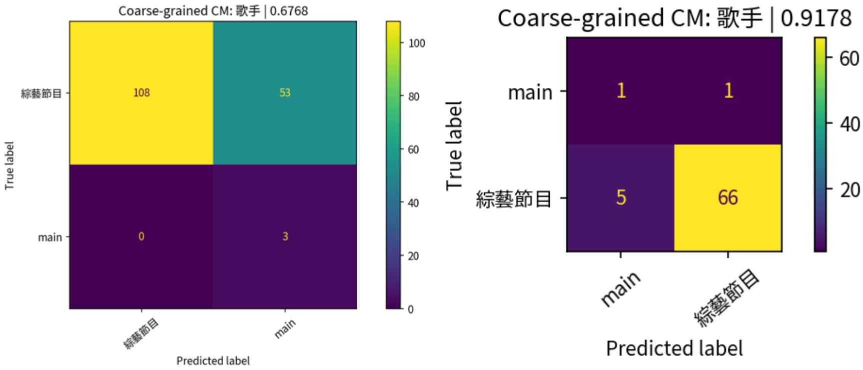


Fig.6. Confusion matrix of “歌手”: TFIDF (left) vs. W&L (right)



V. Conclusion

This project is inspired by the Wiki-based WSD systems experimented in several previous works, and it is shown that both TFIDF and word2vec & Lesk++ perform quite well especially when coarse-grained sense count is limited under 6. Lesk++ in particular, with its simple implementation and intuitive mathematics, proves to be a good alternative for knowledge-based supervised WSD systems. For future works, the dataset in the project is extracted from Wikipedia and cleaned by heuristics, which could contain some flaws (eg. incorrect gloss boundaries, senses that do not have an established page) that could be removed by manual review. It helps if the dataset could be cleaned and experimented again with strictness. Furthermore, the Wikipedia words’ senses are usually arbitrarily defined or defined only for proper nouns, and therefore it is hard to generalized to corpora that do not define senses in Wikipedia’s way, also some senses are lost too. It might be interesting to d map the senses to WordNet sense, so that the sense-files are tagged in WordNet and are more likely to benefit future WordNet-relevant applications.

VI. References

Dandala, B. & Mihalcea, R. & Bunescu, R. (2013). Word Sense Disambiguation Using Wikipedia. https://aclanthology.org/I13-1057.pdf.

Oele, D., & Noord, G.V. (2018). Simple Embedding-Based Word Sense Disambiguation. GWC. https://www.semanticscholar.org/paper/Simple-Embedding-Based-Word-Sense-Disambiguation-Oele-Noord/25f93681a329787831d3fcf9d268f8820b0deaf6.

Sara, T., Claudio G., & Kateryna, T. (2013). Wikipedia-based WSD for multilingual frame annotation. *Artificial Intelligence*, 194, 203–21. https://www.sciencedirect.com/science/article/pii/S0004370212000720?via%3Dihub.

VI. Appendix

Appendix A:

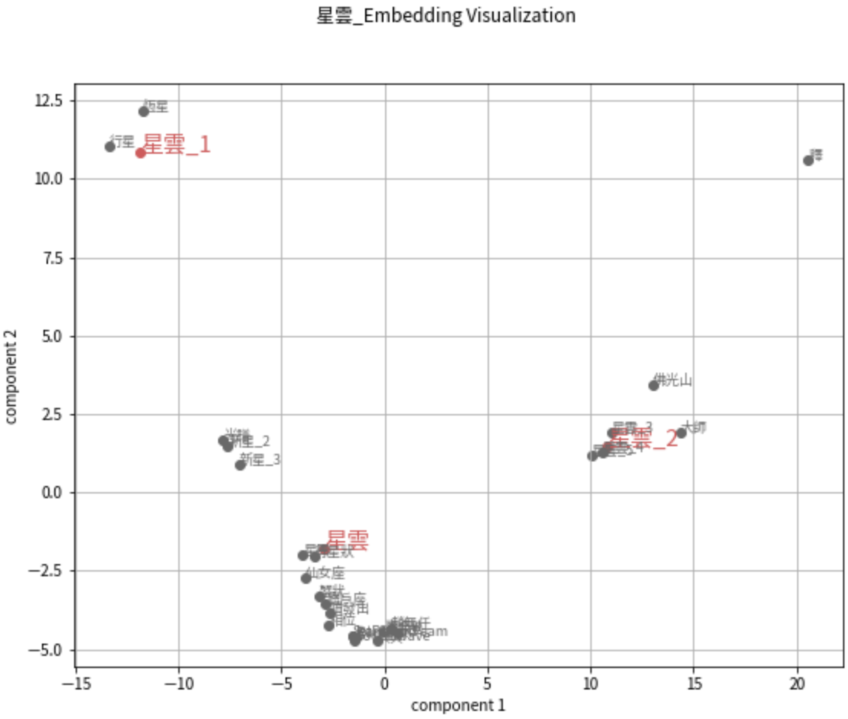
The selected “星雲” embedding visualization of word2vec sense embeddings.

The sense-index to gloss mapping is:

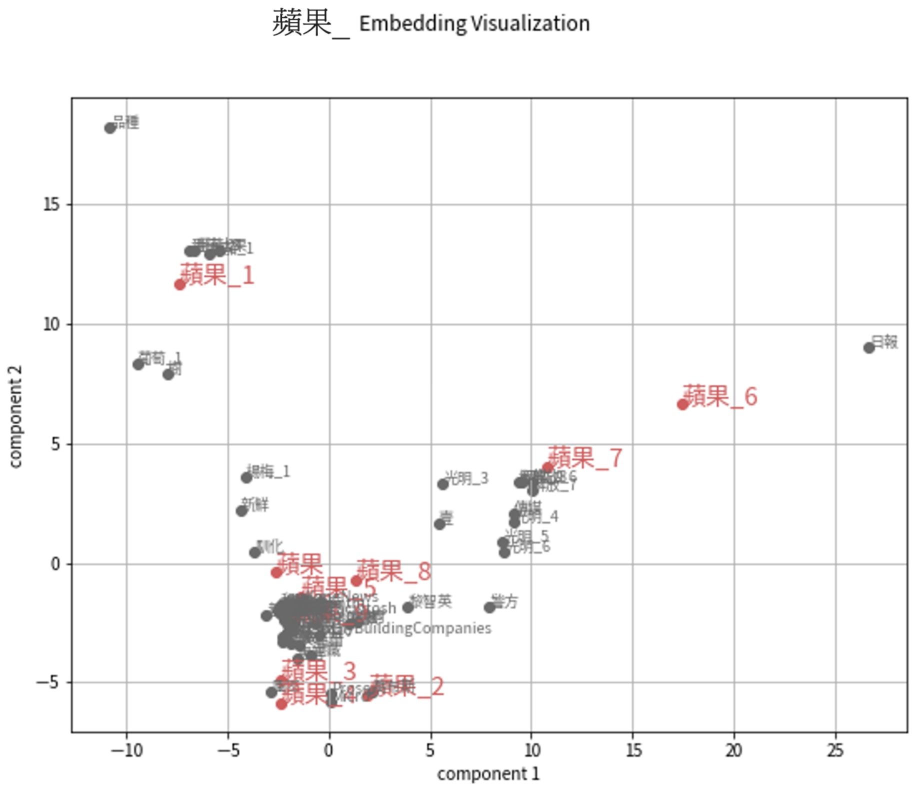
星雲: the un-disambiguated embedding for “星雲” in other sense-files.

星雲_1: 星雲，是一種星際雲、星際物質。

星雲_2: 「星雲大師」，1927 年生，國際佛光會的創辦人。



Appendix B: The “蘋果” embedding visualization of word2vec sense embeddings.



The sense-index to gloss mapping is:

蘋果: the un-disambiguated embedding for “蘋果” in other sense-files.

蘋果_1: 蘋果樹（學名：Malus domestica）是薔薇科蘋果亞科蘋果屬植物，為落葉喬木

蘋果_2: 蘋果公司，著名電子產品生產商。

蘋果_3: 蘋果園區，蘋果公司於 2017 年 4 月起啓用的公司總部新址。

蘋果_4: 蘋果唱片公司（英語：Apple Corps），披頭士樂團創立的唱片公司

蘋果_5: 蘋果日報。

蘋果_6: 蘋果日報 (香港)，香港公司壹傳媒在香港發行的報紙。

蘋果_7: 蘋果日報 (臺灣)，香港公司壹傳媒在臺灣發行的報紙。

蘋果_8: 蘋果日報慈善基金，香港一個慈善基金，由壹傳媒有限公司於 1995 年成立。

蘋果_9: 蘋果 (電影)，2007 年上映的中國電影。

Note that the 光明_ embeddings near 蘋果_6, 蘋果_7 belong to a news agency and its websites. Note also that the un-disambiguated embedding isn’t around the main sense 蘋果_1, but around the tech company Apple Inc., meaning that most of the “蘋果” shown in other sense files refer to, or at least are semantically closer to the sense of Apple Inc..