Automatically Suggesting Example Sentences of Near-Synonyms for Language Learners

Chieh-Yang Huang Nicole Peinelt Lun-Wei Ku
Institute of Information Science,
Academia Sinica, Taipei, Taiwan.
appleternity@iis.sinica.edu.tw, nis50122806@gmail.com, lwku@iis.sinica.edu.tw

Abstract

In this paper, we propose GiveMeExample that ranks example sentences according to their capacity of demonstrating the differences among English and Chinese near-synonyms for language learners. The difficulty of the example sentences is automatically detected. Furthermore, the usage models of the near-synonyms are built by the GMM and Bi-LSTM models to suggest the best elaborative sentences. Experiments show the good performance both in the fill-in-the-blank test and on the manually labeled gold data, that is, the built models can select the appropriate words for the given context and vice versa.

1 Introduction

Integrating new words into active vocabulary requires language learners to make connections between the new lexical items and their previous knowledge. The acquisition of (near-)synonyms is especially challenging as learners need to know in which respects the words are similar and in which ways they differ from each other in order to make correct lexical choices while composing sentences. While absolute synonymy, i.e., interchangeability of words in any context, is generally a rare linguistic phenomenon, near-synonyms, which are similar words that differ in mostly only one aspect, are relatively common and often confuse learners with small nuances between them (DiMarco et al., 1993).

In order to assist language learners with the acquisition of near-synonyms, we previously developed GiveMeExample (Chieh-Yang and Lun-Wei, 2016) , a system that allows users to search for a pair of similar words and obtain a number of ranked example sentences which best highlight the difference between the words. Based on these examples, learners can derive the different usage patterns. In this paper, we introduce the enhanced edition, including the basic functions with the added language support for Chinese, an automatic difficulty scorer, an improved word usage model and a visualization feature for all the ranked example sentences. The online GiveMeExample system is available at http://givemeexample.com/GiveMeExample/.

2 Example Sentence Suggestion

GiveMeExample recommends useful and clear example sentences in two stages: first filtering out complicated sentences by the **automatic difficulty scorer** and then ranking the remained sentences by their **clarification ability**, which indicates the capability of a sentence to clear up confusion of words. We start from introducing the materials for the system design.

Experimental Material and Automatic Difficulty Scorer For the English version, the sentence pool is assembled using example sentences from Vocabulary.com[1]. On the other hand, the Chinese pool is composed of sentences collected from two balanced corpora, the Lancaster Corpus of Mandarin Chinese (McEnery and Xiao, 2003) and the UCLA Written Chinese Corpus (Tao and Xiao, 2007). All of the

[1]Text from Vocabulary.com (https://www.vocabulary.com), Copyright ©1998–2016 Thinkmap, Inc. All rights reserved.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations, pages 302–306, Osaka, Japan, December 11-17 2016.

sentences here are utilized to train the word usage model, but in order to provide useful example sentences for learners, we further filter out complicated sentences and build with remaining sentences the simple sentence pool, from which the final example sentences are chosen.

To filter out complicated sentences, we build the automatic difficulty scorer based on the work of Pilán et al. (2014) but with several modifications. First, in order to assign to each sentence a score (as opposed to a class) we use a linear regression model instead of SVM. Second, the Swedish-specific features introduced by Pilán et al. (2014) are omitted. The training data for English sentence difficulty scorer is manually labeled by a native speaker who grades sentences from two aspects, the difficulty of wording and the complexity of sentence structure, ranging from 1 to 4. The sentence difficulty score ranging from 2 to 8 is then obtained by summing up these two scores. However, for the Chinese sentence difficulty scorer, the training data is collected from mock tests for Hanyu Shuiping Kaoshi (HSK), a Chinese Proficiency Test, and the difficulty degree of a extracted sentence corresponds to the proficiency degree of the content that this sentence comes from.

Measuring Clarification Ability of Sentences When searching for useful example sentences for the target word w_i in a word confusion set W, there are two related factors: (1) **Fitness**: the probability $P(s|w_i)$, whether w_i is appropriate for the example sentence s given a slot to put w_i. $P(s|w_i)$ is calculated by the word usage model. (2) **Relative Closeness**: the summation of the differences of between probabilities $P(s|w_i)$ and $P(s|w_j)$, i.e., $\sum_{w_j \in W - w_i} P(s|w_i) - P(s|w_j)$. A high relative closeness score denotes a better fit of s to w_i and a worse fit to $W - w_i$. We then calculate the clarification score with the multiplication of the fitness score and the relative closeness score:

$$score(s|w_i) = P(s|w_i) * (\sum_{w_j \in W - w_i} P(s|w_i) - P(s|w_j)) \quad (1)$$

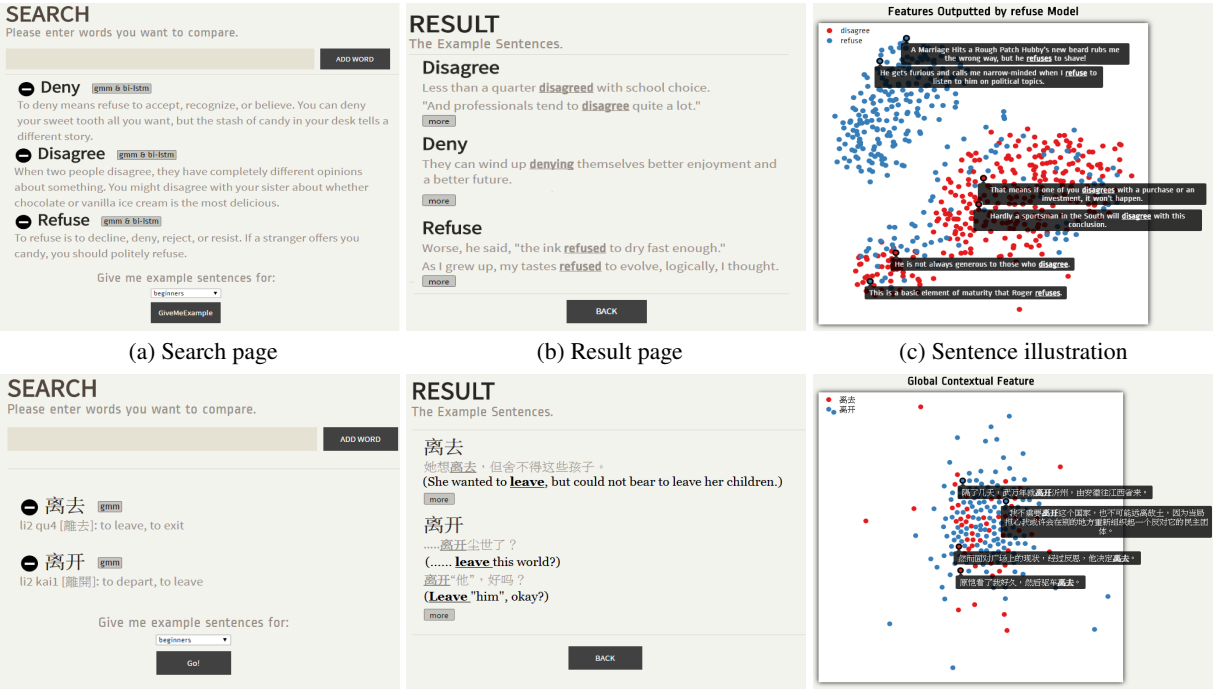
where $score(s|w_i)$ denotes the clarification score of the example sentence s for w_i. We generate recommendations by ranking sentences in the simple sentence pool by their clarification scores. Then we repeat this procedure for all words in word confusion set W to find their elaborative example sentences. Next, we describe the calculation of the probability $P(s|w_i)$.

Word Usage Model To estimate $P(s|w)$ for an observed sentence s, we build a word usage model for the word w. The word usage model is built as an one-class classifier to recognize target samples from an unknown sample space and to process dynamically requested word confusion sets without retraining the models. We introduce two word usage models, the Gaussian Mixture Model (GMM) (Xu and Jordan, 1996) with contextual feature and the Bi-directional Long Short Term Memory neural network (Bi-LSTM) (Graves et al., 2013; Schuster and Paliwal, 1997; Hochreiter and Schmidhuber, 1997).

To build the GMM model, for each sentence $s = w_1 \cdots w_{t-k} \cdots w_t \cdots w_{t+k} \cdots w_n$, where w_t is the target word and k is the window size, we take the k words preceding and following the target word and represent them as well as their adjacent combinations in sequence using the summation of their word embeddings (Pennington et al., 2014). For example, the feature extracted by the windows size $k = 2$ is $\{e_{w_{i-2}} e_{w_{i-1}} e_{w_{i-2,i-1}} e_{w_{i+1}} e_{w_{i+2}} e_{w_{i+1,i+2}}\}$, where e_w denotes the summation of word embeddings of word sequence w. Next, GMM applies Expectation–Maximization algorithm to estimate its parameters and approximate to the data distribution. Empirically, we find that the GMM model with $k = 2$ and $number$ (of mixture) $= 50$ achieves the best performance. To train the GMM model for the target word w_t, a total of 5,000 corresponding sentences are used as the training samples.

To build the Bi-LSTM model, following the same idea of using contextual features, we take words adjacent to the target word into account. However, rather than using the information limited in a small window, Bi-LSTM exploits all the preceding and the following words of the target word in the sentence by a forward LSTM and a backward LSTM respectively. Then the output vectors of these two LSTM are concatenated together to form the sentence embedding, which is also a kind of contextual feature of the given sentence. At last, we add two fully connected layers as the binary classifier to predict whether w_t is appropriate for s. To train a Bi-LSTM model for w_t, we use 5,000 sentences containing w_t as the positive samples. For the negative samples, we randomly choose another 50,000 sentences (10 times of the positive samples) which do not contain w_t. In the end, a total of 55,000 sentences are used to train the Bi-LSTM model for each w_t.

3 System



(a) Search page (b) Result page (c) Sentence illustration

(d) Chinese version search page (e) Chinese version result page (f) Sentence illustration

Figure 1: The user interface of `GiveMeExample`

Near-Synonym Set Search Interface Fig. 1a and Fig. 1d show the search interface. On the search page, learners can type in the words they want to compare easily and dynamically. In addition, `GiveMeExample` also offers a brief explanation of each word. The "gmm &bi-lstm" flag on the right side of the searched word indicates that `GiveMeExample` can provide example sentences suggested by both two models. Furthermore, learners can adjust the difficulty level with a drop-down list to find example sentences according to their language proficiency.

Fig. 1b and Fig. 1e show the result interface where the system-suggested example sentences are listed. `GiveMeExample` provides a "more" button to retrieve additional elaborative sentences. This function facilitates learners to reach more example sentences to generalize the usage and make inference about their difference. In Fig. 1b, learners can conclude that only "refuse" is followed by "to Verb" but the other two words are not. However, example sentences of "disagree" and "deny" do not demonstrate explicit usage. As a result, more example sentences are needed for learners to infer the correct usage of these two words. A similar situation occurs in Fig. 1e. Learners can conclude that only "离开" can be followed by an object, but "离去" cannot. However, for "离去", learners may need more example sentences.

Example Sentence Illustration In the example sentence illustration page, each sentence is turned into a two dimensional point by applying t-Distributed Stochastic Neighbor Embedding (t-SNE) (Maaten and Hinton, 2008) on its contextual feature vector (either from the GMM or Bi-LSTM model). When Bi-LSTM model is used, we will generate one figure for each word because each Bi-LSTM word usage model has it own feature extractor. One confusing word will have different feature vector in different word models. Therefore, for a confusion word set, we will have several figures for one confusing word from several word models. Fig. 1c illustrates the figure on the near-synonym set {refuse, disagree} generated by the Bi-LSTM word usage model of "refuse". `GiveMeExample` displays the sentence represented by each point when the user hovers the mouse over it. With this function, learners can easily search for either different or similar usage patterns of a set of confusing words. For instance, in Fig. 1c, the top two blue points of "refuse" both show the usage "someone refuses to do something" of this word and thus are grouped together, while the red-blue mixed small group at the lower left corner of the figure showing similar usage "someone refuses/disagrees" of the two confusing words "refuse" and "disagree".

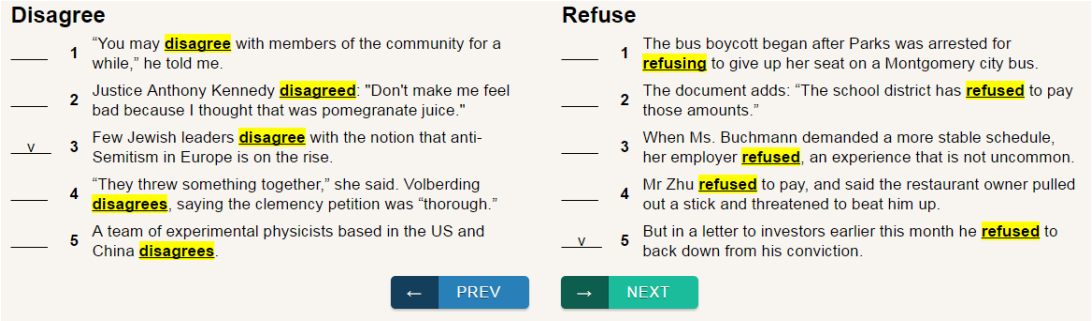


Figure 2: An example question for learners. Learners need to select the best sentence pair, one from the left column and one from the right, to best illustrate the difference between {refuse, disagree}.

In this figure learners can easily find that the example sentences are roughly grouped into three clusters, suggesting in general there are three major usages of these two words.

Differing Aspect	Near-synonym Pair	Score	Differing Aspect	Near-synonym Pair	Score
abstract vs. concrete	blunder - error	7/10	low vs. high degree	mist - fog	2/10
	维护- 保护	6/10		经常-往往	3/10
formal vs. informal	child - kid	6/10	pejorative vs. favorable	skinny - slim	3/10
	购买- 买	9/10		产生-造成	8/10

Table 1: How useful the recommended example sentences can help discriminate near-synonyms.

Evaluation We first evaluate Fitness by the FITB test, which assesses whether the proposed fitness score can identify the appropriate context for a given word. A FITB question contains a sentence with a blank field to be filled in by several near-synonym candidate answers. We adopt Edmonds benchmark for evaluation. Edmonds (1997) suggests the FITB test on 1987 Wall Street Journal (WSJ) and defines 7 near-synonym sets, and after that it becomes a benchmark. Among the one-class models ever reported in the literature, the 5-gram language model is the best (acc 69.90%) (Islam and Inkpen, 2010). However, results show our proposed GMM and Bi-LSTM both outperform it by achieving the accuracy 70.26% and 73.05%, respectively.

Then we evaluate Clarification by the Learner Glod standard (LG) experiment. We define 10 near-synonym sets, where each set contains 2 near-synonym verbs, for evaluation. For each near-synonym set we build 20 questions. Fig. 2 shows an example question for the word confusion set {refuse, disagree}. Each question contains 5 randomly chosen example sentences for each confusing word. The sentences are listed in parallel, 5-to-5, in each question and 6 learners are requested to choose the best sentence pair, one sentence for each word. The pair selected by learners are treated as the gold answer and thus each question has at most six gold pairs from learners. Then the `GiveMeExample` system answers each question by regarding it as question of 25 sentence-pair choices. The sytem will rank all these 25 choices and among them the rank of best gold pair are used to calculate the mean reciprocal rank (MRR). The MRR of GMM and Bi-LSTM are 0.502, 0.500 respectively, and both outperform the random-ordered baseline (0.423) and first-seen baseline(0.429).

Discussion In order to investigate how helpful the proposed system is, we conduct a case study for a number of Chinese and English near-synonym pairs that differ in certain linguistic aspects (abstract vs. concrete, formal vs. informal, low vs. high degree and pejorative vs. favorable) as proposed by DiMarco et al. (1993). The ten highest ranked sentences from the system are manually scored for their suitability to discriminate each two confusing words, conferring one point on a good example sentence and zero points if the sentence did not highlight a difference between the two near-synonyms. The main criterion we employ for this decision is whether the synonyms are mutually exchangeable without altering the meaning and normality of the original statement (Cruse, 1986). According to our results, the system makes good suggestions for the abstract vs. concrete and formal vs. informal word pairs (Table 1) with 60 to 90 percent of helpful example sentences. Especially the formal and informal difference is the most recognizable from the example sentences, e.g. “Would you let your kids smoke pot?” vs. “The New York City Children’s Chorus will perform during the worship service.” Results for the pejorative vs. favorable

word pairs are mixed, while the suggested sentences for the near-synonyms with varying degrees are found to be less distinctive by their associated context. The difference in performance seems largely related to the extend to which the words differ from each other, making "mist - fog" more difficult to distinguish than "error - blunder".

The analysis of the recommended sentences also shows that the system picks up two additional aspects in which near-synonyms may differ. First, many suggested sentences contain collocational patterns which are helpful to distinguish similar words, such as fixed expressions e.g. "error bars" but not "blunder bars" and common arguments e.g. "维护 平衡" vs. "保护 鸟种". Second, the system can demonstrate monofunctional vs. polyfunctional properties of near-synonyms, e.g. "error" can only serve as noun, but "blunder" can function as a noun ("What a jolly blunder Police Headquarters would make!") as well as a verb ("Ye blundering idiot!").

4 Conclusion

We have proposed the `GiveMeExample` system to help language learners understand English and Chinese near-synonyms by utilizing learners' ability to learn implicitly from the comparison of good example sentences. We have shown that `GiveMeExample` has the design to support the online comparison of arbitrary words and can perform satisfactory. In the future, we plan to support the analysis of phrases and further investigate the effect from the learning side.

Acknowledgements

Research of this paper was partially supported by Ministry of Science and Technology, Taiwan, under the contract MOST 104-2221-E-001-024-MY2.

References

Huang Chieh-Yang and Ku Lun-Wei. 2016. Givemeexample: Learning confusing words by example sentences.

D Alan Cruse. 1986. *Lexical semantics*. Cambridge University Press.

Chrysanne DiMarco, Graeme Hirst, and Manfred Stede. 1993. The semantic and stylistic differentiation of synonyms and near-synonyms. In *AAAI Spring Symposium on Building Lexicons for Machine Translation*, pages 114–121.

Philip Edmonds. 1997. Choosing the word most typical in context using a lexical co-occurrence network. In *Proceedings of EACL 1997*, pages 507–509. Association for Computational Linguistics.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE ICASSP*, pages 6645–6649. IEEE.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Aminul Islam and Diana Inkpen. 2010. Near-synonym choice using a 5-gram language model. *Research in Computing Sciences*, 46:41–52.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Tony McEnery and Richard Xiao. 2003. The lancaster corpus of mandarin chinese.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Ildikó Pilán, Elena Volodina, and Richard Johansson. 2014. Rule-based and machine learning approaches for second language sentence-level readability. In *BEA Workshop 2014*, pages 174–184.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Hongyin Tao and Richard Xiao. 2007. The ucla chinese corpus (2nd edition).

Lei Xu and Michael I Jordan. 1996. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151.