

# Statistics and Machine Learning 21:198:329

## Preposition learning final project

Libby Barak

April 9, 2020

Second language speakers (L2) struggle to learn the conventional ways of language use in their second language. Among the hardest linguistic properties to learn in L2 is proper use of prepositions, such as, *for*, *to*, and *in*. In this project you will model the knowledge of prepositions as a KNN clustering problem. The goal of the project is to compare three language models, English as a native speaker (English-L1), Mandarin as a native language (Mandarin-L1), and Mandarin as native language with English as a second language (Mandarin-English-L2). The three models will be trained on data that represents their learning scenario and will be compared on the ability to choose the correct preposition for an English sentence.

To do, you will implement the following steps:

1. Pre-process training sentences.
2. Pre-process testing sentences.
3. Extract and combine semantic vector representation for training and testing data.
4. Train a KNN model for three language models using the training sentences (English-L1, Mandarin-L1, and Mandarin-English-L2).
5. Test the model by predicting the correct preposition for a set of test sentences.
6. Report your results using statistical analysis methods.

You will be provided with 4 files:

1. Sentence for English training
2. Sentence for Mandarin training (Mandarin data will be provided as English text already translated).
3. Word2Vec embeddings to represent the words and sentences in the data.
4. Test data - Multiple choice questions with the correct choice and choice-probability based on a human-subject study.

Table 1: Examples of sentences in English and in translated data

English
<i>det:dem that adj big n machine pro:rel that pro:sub we v see&amp;PAST adv out prep on det:art the n street .</i>
<i>pro:int whatpost elseaux be&amp;3Sprep indet : art then book?</i>
<i>v lookprep atdet : poss hisn toe-PL</i>
Mandarin
<i>pro wo3 - PL=Iprep zai4=atn jia1=familyv zhao4xiang4=take_a_picture sfp a1 .</i>
<i>pro wo3=Iv na2=holdprep gei3=forpro ni3=youv kan4=look.</i>

The data for the project is obtained through research agreement. It is protect with privacy agreements and cannot be latter used for used for industrial or personal reasons. You can use any code written in the course to complete the project. You are allow to use any code freely available through web-search to assist you with any of the steps.

## 1 Training data

The data for training is taken from CHILDES database (MacWhinney & Snow, 1990). CHILDES includes child-directed speech recorded, transcribed and annotated using data from caregivers, researchers, and children interacting in naturalistic settings. The data for training has the following form presented in Table 1.

In both language types, each sentence is written in a single line. Sentences are separated by an empty line. Words in a sentence are separated by a space, or a ‘ ’. The words in the sentence are given with their part-of-speech and the lemmatized form. The part-of-speech indicate the syntactic role of the word in the sentence. The lemmatize form removes any inflections from the word to get a canonical form, e.g. *saw*, *seen*, *seeing* would all be lemmatized into *see*. The part-of-speech is separated from the words using ‘|’ delimiter. The word may contain lexical information separated by ‘&’ or ‘-’, and in Mandarin, the translation separated with a ‘=’. For example, *n|toe-PL* stands for *toes*, *v|see&PAST* stands for *saw*, and *v|zhao4xiang4=take\_a\_picture* stands for *take a picture*.

Each sentence represents one input item for a specific prepositions. For training, you will need to identify the preposition represented by the sentence and extract all the relevant words. Notice that a sentence can include more than one preposition. You can identify the prepositions since they start with ‘prep’ tag. You will need to use each sentence as a training example for all the prepositions in it. The cleaning process includes:

1. Identify all preposition in the sentence
2. Extract the individual words from the sentence using the delimiters ‘ ‘, and ‘ ’.
3. From the list of extracted words, remove words with any of the following parts-of-speech: ‘pro:rel’, ‘co’, ‘det:art’, ‘det:poss’, ‘neg’, ‘aux’, ‘mod’, ‘cop’, ‘cl’, and ‘cm’ (Think why this tags are not helpful in training).

4. Clean the words removing the part-of-speech tag, translation, and lexical information.
  - Remove part-of-speech using the ‘|’ delimiter.
  - Remove translation (if any) using the ‘=’ delimiter.
  - Remove lexical information (if any) using ‘&’ or ‘-’ delimiters.
  - Break into individual words if contains ‘\_’.
5. For each preposition create a list of the words within a 4 words window of the preposition, without the preposition itself (4 words before and 4 words after the preposition).

At the end of this process, you should have for each preposition in the data, lists of 8 or less words representing each training item for the preposition. For example, the sentence “*det:dem|that adj|big n|machine pro:rel|that pro:sub|we v|see&PAST adv|outprep|on det:art|then|street*” should result in an input item for the preposition *on* including: {‘machine’, ‘we’, ‘see’, ‘out’, ‘street’}.

## 2 Vector representation

Once you extracted the training items for each preposition, you need to convert the list of words to a vector representation. The vector representation will allow you to use the sentence to compute similarity between sentences. For this part, you will use vectors created for each words. The vectors were create by training the Word2Vec model on all the texts from the English Wikipedia dump (October 2019)<sup>1</sup> (Kutuzov, Fares, Oepen, & Velldal, 2017). The first line in the file can be skipped (contains the number of words in the file and number of dimensions in each vector representation). Each of the following lines corresponds to a single word following the format: “*first\_ADJ* 0.005799 0.024848 ...”. Each line contains 301 parts separated by a space. The first part is the word itself followed by the part-of-speech separated by a ‘\_’. For this part, you can ignore the part-of-speech data.

In this step, your goal is to take the list of words created in the previous steps and obtain a single embedding vector. An embedding vector is a sequence of numbers representing the valency of the meaning of the word over dimensions of meaning, in our case, 300 dimensions. You should get the vectors using the following steps:

1. Create a list of all the words you require for the training data
2. Find the corresponding lines in the embedding vectors file. If the embedding vector file contains more than one line for a word, take the first occurrence. If the word does not appear in the vectors file, skip the word.
3. For each training item, create a single vector by summing the vectors for all the words. The vectors are summed by dimension, i.e., summing the 1st dimension of all words, 2nd dimension, and so on. The resulting vector will be a 300 dimension vector like the vectors for the words.

You now have the mathematical representation of each training item needed to train a KNN model. You can create a model by training a cluster for each of the prepositions in the data.

0.170454545 0.693181818 0.068181818 0.068181818

---

<sup>1</sup>(See <http://vectors.nlp1.eu/explore/embeddings/en/models/>)

Question	Choice	Preposition	Correct	Participants
She has a low opinion ___ herself	A	with	0	0.17
She has a low opinion ___ herself	B	by	0	0.69*
She has a low opinion ___ herself	C	of	1	0.07
She has a low opinion ___ herself	D	inside	0	0.07
Max plugs himself ___ again	A	as	0	0.10
Max plugs himself ___ again	B	on	0	0.19
Max plugs himself ___ again	C	among	0	0.39*
Max plugs himself ___ again	D	in	1	0.32

Table 2: Examples of test items included in the experimental design of (Xiao et al., 2018). Each question has 4 possible answers with specific prepositions. Correct choice marked by value 1 in the correct column. The percent of participants choosing each prepositions is given in the Participants column. The preposition with the highest value in the Participants column marks the top choice for the question (marked with an asterisk).

### 3 Testing data

The testing data is based on multiple choice questions answered by native speakers of Mandarin. Participants choose a preposition from 4 options as a filler for a given sentence. See Table 2 for an example. We would like to test the model by comparing its prediction for each of the 4 options to the choices made by participants. The testing data file contain 25 questions with 4 choices per question.

You should clean up the sentences similar to how you cleaned the training sentences. There is no part-of-speech tagging or lemmatization for these sentences. You can use an online interface to help you do this (e.g., <http://textanalysisonline.com/spacy-word-lemmatize> for lemmatization and <https://text-processing.com/demo/tag/> for part-of-speech tags). Remove any word that was removed in the training sentences, or if you use the online interface, remove words corresponding to ADV, CONJ, DET, and POS. Remember that these are only ten sentence, the online interfaces for preprocessing should be easy and faster to use compared with writing code for this step.

Use the clusters created in the previous step to estimate how good each of the 4 prepositions for the sentence. For example, The first sentence from Table 2 should be represented by the sum of the vectors corresponding to the words {'she', 'opinion', 'herself'}. For each of the training items in the cluster for 'with', the similarity score for the training sentence and the test sentence can be calculated using cosine similarity:

$$\cos(t, q) = \frac{t \cdot q}{||t|| \cdot ||q||} = \frac{\sum_{i=1}^{300} t_i q_i}{\sqrt{\sum_{i=1}^{300} (t_i)^2} \sqrt{\sum_{i=1}^{300} (q_i)^2}} \quad (1)$$

where  $t$  is a training sentence,  $q$  is a question sentence, and  $t_i$  and  $q_i$  correspond to the  $i$ -th dimension of their vector representation. Cosine similarity provides a score between 0 and 1 for each pair of training and testing items. Now, we can use the equation provided in class to choose the top N items in the cluster to estimate the likelihood of this preposition as the correct choice for the question. This step should be repeated for each of 4 preposition choices for the question. The

final score for each of the 4 choices for each of the question is obtained by normalizing the scores given each of the 4 clusters:

$$P_B(c|q, D) = \frac{s_c + 1}{\sum_c s_c + 4},$$

where,  $c$  is a choice for the question  $q$ ,  $s_c$  is the score the model predicted based on all the training data  $D$ . This normalization step makes sure the scores for the 4 choices sum to 1 to align with the human data. We add 1 to the nominator and 4 to the denominator to account for the possibility that a certain choice has 0 probability according to the model (e.g., because the preposition was not captured by the data).

## 4 Evaluation

Training, getting Word2Vec vectors, and Testing, should be repeated 3 times to get 3 models. The first model using the English data only, the second model using the Mandarin data only, and the third model using the Mandarin data and the first half of the English data. The results from each of the three models should be added to the testing data file as a new column with a score for each preposition.

You should report the correlation of each of the models with the participants choice for all 25 questions provided. You should include 3 correlation score in your report:

1. The correlation of the scores with all choices (100 values).
2. The correlation of the scores with the correct choices for each question (25 values).
3. The correlation of the scores with the top choice for each question (the choice made by most of the participants).

You can use the slides from the talk as reference. Please provide an analysis of the results. Look at the data and see which words were selected to represent each sentence, which sentences were predicted correctly by the model (top choice of the model was the correct choice), which model predicted the top choice correctly (top choice of the mode was also the top choice by the participants). Based on your impression, your understanding of the project, and the correlation results, write in a short paragraph what are some of the advantages and shortcomings of this modeling approach. Note that there are no right and wrong answers to this. I am simply asking for your opinion.

Good luck!

## References

- Kutuzov, A., Fares, M., Oepen, S., & Velldal, E. (2017). Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *Proceedings of the 58th conference on simulation and modelling* (pp. 271–276).
- MacWhinney, B., & Snow, C. (1990). The child language data exchange system: An update. *Journal of child language*, 17(2), 457–472.
- Xiao, W., Wang, M., Zhang, C., Tan, Y., & Chen, Z. (2018). Automatic generation of multiple-choice items for prepositions based on word2vec. In *International conference of pioneering computer scientists, engineers and educators*. Springer.