Extensions of Cross-Lingual Embeddings for Bilingual Dictionary Induction

Andrew Martin
University of Pennsylvania
andrmar@sas.upenn.edu

Shivansh Inamdar
University of Pennsylvania
shivansh@sas.upenn.edu

Abstract

We explore extensions to projective methods in creating cross-lingual embeddings. The problem of using cross-lingual embeddings to predict missing entries in a bilingual dictionary is resemblant of translation and we try our hand at this task using three main extensions of an orthogonal projection method: "sewing" language spaces together, finding a linear transformation between spaces without a supervising dictionary, and using a second projection to close the space between a word and it's translation. Our results performed mostly at about the same level as our published baseline with no significant improvement. In some cases such as not using a supervising dictionary, we performed somewhat worse, although still pretty well given the lack of supervision.

1 Introduction

The problem we chose was to explore robust models for Cross-lingual word vector embeddings. While there is extensive work on monolingual word embeddings, cross-lingual embeddings do not seem to garner as much interest despite their extreme relevancy for task such as Bilingual Dictionary/Lexicon Induction, Word Similarity, and Cross-Lingual Hypernym Discovery.

Monolingual embeddings are typically trained with one of two methods, SKipGram and CBOW. SkipGram trains classifiers to predict the probability that all words in a given window (around the focus word) would appear in that window. CBOW uses the words in the window to predict the word with the highest probability to be in the focus of the window/bag. SkipGram takes as input a focus word and makes classifiers for the words in the window while CBOW takes a window and guesses the focus word.

Recently, there has been a trench in projection based cross-lingual models (some of which we will mention/explore here), in which given two vector spaces A and B, the goal is to project both spaces onto a third common vector space in order to take advantage of semantic similarities in the spaces. Usually this is done with an assumption that the two spaces are isomorphic so that these semantic properties will be reserved but it is often the case that two spaces are not isomorphic due to intricacies like gender-ed words in languages likes Spanish and French. Once the isomorphic assumption has been made, orthogonal transformations are applied to the vector spaces to bring them together.

The task at hand that we chose to tackle is that of Bilingual Dictionary Induction. That is, given a bilingual dictionary with missing entries, we would like to fill in those entries with a prediction for the translation. For example given the word "surprised", we might want to predict "tonn" or "surpris".

We chose this project to help dive deeper into cross-lingual embeddings and study semantic similarities between languages through their embeddings. Afterall, most languages fall into one of a few major language families and many derive similarities from this shared ancestry. It is interesting to explore how languages function similarly across cultures and linguistic divides.

2 Literature Review

"Improving Cross-Lingual Word Embeddings by Meeting in the Middle", Doval et. al (2018):

A type of model that has become recently

are those that involve projections of embeddings onto a smaller subspace. The idea behind the this paper was to take monolingual embeddings and align their respective vector spaces through linear transformations using a small bilingual dictionary for the supervising component and then to map each word embeddings onto its average value.
The inspiration behind this is that the assumption that two monolingual spaces have identical structures is not a good one. Using the assumption of the two isomorphic structures leads to using an orthogonal component in the linear transformation between the two vector spaces. However, dropping this requirement can lead to overfitting.
Instead, for each word w with translation w, the authors would solve for a mapping that sends the word w to the average of w and w, thereby creating a cross-lingual vector space that can be intuitively thought of as the average space.
For the original linear transformation, the authors tested VECMAP and MUSE methods, and then added the additional projective mapping. The languages they tested on included English, Spanish, Italian, German, and Finnish, using web-extracted corpora. Additionally, the original monolingual embeddings were trained with the skipgram model from fastText.
They tested on the tasks of bilingual dictionary induction, cross-lingual hypernym discovery, and word similarity. For the first, the suggested model beat all other models (except on Finnish), for the second task, there was minimal improvement, and for the final task, the new model surpassed the older models using larger data training data.

"Cross-lingual models of word embeddings: An Empirical Comparison", Upadhyay et. al (2016):

The goal for this paper was to empirically compare four cross-lingual word embedding models that each required different form of alignment(s) as supervision. The tests were done using four different language pairs: Eng-German, Eng-French, Eng-Swedish, Eng-Chinese. In general, they remark that word vectors trained with expensive cross-lingual supervision performed best on semantic tasks, while weaker supervised ones do better on syntactic tasks. They looked at dense, fixed length (200 dim) embeddings trained using: BiSkip, BiCVM, BiCCA, BiVCD. The final parameters chosen for the report were the ones that led to the best overall performance of the model.
On monolingual word similarity BiCVM performed best, for dictionary induction BiSkip won, for document classification BiSkip won, for QVEC tasks, BiSkip won overall, though BiVCD won for Swedish to English translation, for syntactic dependency parsing BiCCA won for the models trained and evaluated on different languages, while BiSkip did comparably to BiCCA (and sometimes better) for models trained and tested on same language.

"How to (Properly) Evaluate Cross-Lingual Word Embeddings: On Strong Baselines, Comparative Analyses, and Some Misconceptions", Glavas et. al (2019):

Cross-lingual embeddings are invaluable for tasks like lexicon induction, document classification, information retrieval, dependency parsing, sequence labeling, and machine translation. The authors also remark on the recent trend in projection based approaches that only need word level supervision in dictionaries. The goal of the paper was to evaluate these projection based models using tasks such as document classification, information retrieval, and natural language inference on 28 different language pairs.
Some main questions the paper wanted to address were: Is BLI performance a good predictor of downstream performance for projection-based CLE models?, and Can unsupervised CLE models indeed outperform their supervised counterparts?.
The models tested included CCA, Procrustes, DLV, RCSLS, MUSE, VECMAP, ICP, and GWA. On bilingual lexicon induction (BLI), they used Croatian, English, Russian, French, German, Italian, Finnish, and Turkish for the languages, and they used 300 dimensional fastText embeddings. They created their supervising dictionaries using Google Translate and used the Precision at k evaluation metrics.
The results showed that VECMAP was the most robust choice of unsupervised models, but PROC-B(bootstrapped procrustes), a supervised model, performed best overall. For natural language inference VECMAP and PROC-B performed the best for unsupervised and supervised models respectively, for document classification VECMAP and RCSLS won, and for information retrieval

ICP and PROC-B won.

3 Experimental Design

3.1 Data

We have used a few different types of data:

1. Pre-trained word vectors for 157 languages from fastText. We have included samples for the French and English word vector files. We will expand to more languages later in the project. The files here are in .vec form and are converted to .magnitude to allow us to work with them using PyMagnitude. The vectors were trained with CBOW and postion weights, have 300 dimensions, character n-grams of length 5, and used a window of size 5.

   https://fasttext.cc/docs/en/crawl-vectors.html

2. A dictionary of data separated into 3 parts for train, dev and test. We obtained the dictionary form FaceBook research's MUSE project.

   https://github.com/facebookresearch/MUSEground-truth-bilingual-dictionaries

The first dataset consisted of rows of a word and coordinates specifying the vector associated to the word.

The second dataset has English-French word pairs that are blank-space separated. These are simple text files.

For example:

English	French
pursuit	poursuite
reserved	réservés
johns	johns
verified	vérifié
proportion	proportions
porter	portier
twins	jumelles

3.2 Evaluation Metric

We ended up slightly changing our baseline. Originally, we decided for a simple accuracy count. That is, given three columns where the first is the word in the dictionary, the second is the actual translation, and the third is our predicted translation, we count the proportion of rows where our prediction (the third column) equals the actual translation from the bilingual dictionary (the second column). This worked well originally, but we later found issue when our model would produce the plural version of the correct translation, or a different gendered form, a slightly different form of a verb, or even a synonym to the correct answer. Clearly, these answers should be marked correct or contribute more in favor of our model's performance.

For example, for the English word surprised, the gold translation was étonné whereas we got surpris, étonné, étonnée, supris as our 4 top outputs in that order. Surpris and étonné happen to be synonyms and étonnée is simply the feminine version of the verb, so we thought these should be marked as correct.

Thus we chose to reformulate our problem. Instead of guessing a single translation, we would predict k translations (for the sake of our experiments we chose k = 5). Thus, we would count a prediction correct if any of the k predictions were equal to the actual translation from the dictionary OR if they fall within a certain edit-distance from the actual answer (this is to account for returning a plural form). We found that this showed our model performing much better and helped reward our model for guessing plurals or synonyms since these were often closely followed by a prediction of the actual word.

Consider the example below for our original metric:

English	French	Prediction	Correct
reserved	réservés	réservé	
twins	jumelles	jumelle	
pursuit	poursuite	poursuite	X

Would give an accuracy of 0.33 Since only one of the predictions match the French translation.

3.3 Simple Baseline

Our simple baseline made use of our pre-trained embeddings and the PyMagnitude library for querying words. Given a word w in some language A, our simple baseline would merely return as prediction the top-k closest words to w in the vector space of the target language. These k words act as our predicted translations. Pymagnitude uses the cosine similarity to determine how close two vectors are, given by:

Extension	Fr->En	En->Fr
Published Baseline	0.7208	0.6807
Sewing Model (alpha = 0.1)	0.3322	0.6773
No Bilingual Dictionary	0.6182	0.5863
Meeting In The Middle	0.4042	0.6807

Table 1: Extension performance

$$cosine(A, B) = \frac{A \cdot B}{||A||||B||}$$

4 Experimental Results

4.1 Published Baseline

Our published baseline was based on the work by Mikolov et al. (2013b) with some improvements from Artexte et al (2016). We implemented the linear transformation from the former, with the additional improvements from the latter that maintained orthogonality for the transformation matrix and also required length normalization of the vectors. This model did quite well with our data and proved to be difficult to beat in terms of accuracy with our extensions.

4.2 Extensions

We used three extensions to our published baseline. The results can be found in the corresponding Table 1 above.

4.2.1 Sewing Together Two Vector Spaces

Given two vectors spaces for languages A and B respectively, the goal is to somehow project these spaces onto a common space OR, combine them. In this case we use a bilingual dictionary as supervision to "sew" these spaces together as follows. Given a word pair (w, w') were w' is the translation in space B of word w in space A, we compute the average of these two words and then move each toward this new common point by an amount determined by a parameter α. In our code we utilized $\alpha = 0.1$, though future work could go into optimizing its value. After "sewing" these spaces together, we utilize the published baseline method to project these new spaces together.

4.2.2 Working with no bilingual dictionary

The motivation for this extension were language pairs that might not have extensive bilingual dictionaries readily available. This model uses words that are present in both the vector spaces to create a linear transformation between them. French and English happen to share a lot of words such as entourage, information and comfortable. Additionally however, the two vector spaces also share nouns such as Washington, Twitter, India, UK which we would expect from monolingual vector spaces of multiple European languages. For this extension, we compared the two monolingual vector spaces and created our linear transformation based only on the words that the two spaces had in common. We got the inspiration for this type of extension from a paper by Artexte et. al (2017) where they worked with a very small bilingual dictionary to supervise the model and induced a new bilingual dictionary with that model. They then fed it back into the model and repeated this until some point of convergence where they ended up with quite a good bilingual dictionary. When examining our own data however, we realised that we could generate quite good results without even using a small bilingual dictionary. Some combination of the two methods, where perhaps the starting dictionary could be these common words across monolingual spaces, would probably do even better with very little supervision.

4.2.3 Meeting in the Middle

Inspired by Doval et. al (2018) this method first performs a projection by solving for a mapping from space A to space B by using a bilingual dictionary as supervision. Then, it attempts to solve for a second such projection to the average of the projected vector and the desired translation vector. In other word, given the pairs of (w, w') from the bilingual dictionary as before, we find mapping X such that the sum of $||Xw - w'||^2$ over all such pairs is minimized. This is done using SVD. Then for (w, w') we calculate the average of w and w' and try to solve for the mapping Z that minimizes $||ZXw - avg(w, w')||^2$. This second projection to the average is the "meeting in the middle" in which we try to close the distance between our first projection and the second. This is under the assumption that the first projection left some notable distance between Xw and w'. This second projection also uses SVD to solve this minimization problem.

4.3 Error analysis

As we said before, our models did well with predicting the plural version of the correct translation or even synonyms. This was our main source of error for a while until we implemented the described changes to the evaluation metric.

Some notable examples of errors in the output of our extensions (since they seemed to share many of the same problems) are names. Given the name "Blake", our models might guess words like "Solomon", "Harrison", "Edwards", or "Evans". Clearly, it has recognized that it should return a name but can only manage to return other names. This is likely because names (for the most part) don't have a translation and should remain the same in both languages. That being said, we could potentially extend our models in the future to recognize a named-entity and to just return the named-entity as the prediction.

Additionally, for words like "manger" which means "to eat", we return "cuisiner", "mang", and "nourriture". Respectively, these mean "to dine", "has eaten", and "food". Clearly we are on the right track but not quite pin-pointing the exact word we should return. Though, all things said, we are in the correct ballpark and a larger k (returned predictions) could likely give us the correct answer. Future extensions might explore how to go the extra step to find the precise word to return, AND to adjust our evaluation metric to account for tenses in verbs. We already made a change for plurality of nouns so this wouldn't be too much of a change to implement.

5 Conclusions

As we can see above in the English to French translation, the "Meeting In The Middle" model matched the published baseline. This likely means that the second projection made negligible difference to the model. Perhaps the first projective mapping we solve for did too good of a job in aligning the word pairs. More work is necessary to determine exactly why this is occurring. We also see that the "Sewing" model perhaps a bit worse than the published baseline. This might be that in combining the language spaces we tamper with some of the semantic meaning of the embeddings. Additionally, we might consider using a larger supervising dictionary to avoid a case where some words are moved away from the rest of the language space, while others aren't. A larger dictionary would help preserve the structure by affecting more words in this way. Finally, the "No Dictionary" model, while under-performing the rest, managed to do well considering the lack of supervision. This could be a place to further extend the model, by trying to improve the ability to create robust embeddings without supervision. This would be useful for languages without extensive data/dictionaries available.

As for our French to English translation, the published baseline and the extension using no bilingual dictionary performed slightly better than when going the other way around. Perhaps this can be attributed to English not having gendered verbs as French does and therefore allowing fewer highly likely choices for a word to be translated to. The sewing model and the meeting in the middle model, however, saw their accuracy drop. For the "Meeting in the Middle" model, perhaps the additional projection is causing a movement away from an initial projection that performs remarkably well. As for the "Sewing" model, we see more evidence that we are perhaps tampering with the semantic meaning of the embeddings when combining the two language spaces. Because of the lack of closely related, gendered words in English, this tampering causes a more drastic effect here.

With our extensions, we were able to match our published baseline with a couple of extensions, and, beyond that, show that it was possible to generate quite good results even when not using a bilingual supervising dictionary. We were able to come up with some novel ideas for our extension and we also took inspiration from previously published work in the area of cross-lingual embeddings. Most of all, we found this deep dive into this area of natural language processing absolutely fascinating and enjoyed working on it.

Acknowledgments

We would like to thank Deniz Beser, Diana Marsala, Jasmine Lee, Nidhi Sridhar, Nina Chang, Maria Kustikova, Shashank Garg, and of course, Jishnu Renugopal, and Professor Callison-Burch for all the great work this semester and for making this project possible.

References

Yerai Doval, Jose Camacho-Collados, Luis Espinosa-Anke, Steven Schockaert. *Improving Cross-Lingual Word Embeddings by Meeting in the Middle*. 11 pages. Association for Computational Linguistics.

Shyam Upadhyay, Manaal Faruqui, Chris Dyer, Dan Roth. 2016. *Cross-lingual Models of Word Embeddings: An Empirical Comparison*. 10 pages. Association for Computational Linguistics.

Goran Glavas, Robert Litschko, Sebastian Ruder, Ivan Vulic. *How to (Properly) Evaluate Cross-Lingual Word Embeddings: On String Baselines, Comparative Analyses, and Some Misconceptions Linguistics*. 14 pages. Association for Computational Linguistics.

Thomas Mikolov et. al. *Exploiting Similarities among Languages for Machine Translation*. 10 pages. Association for Computational Linguistics.

Mike Artrtxe et. al. *Learning principled bilingual mappings of word embeddings while preserving monolingual invariance*. 6 pages. Association for Computational Linguistics.

Links to Presentation and Code

Code:
https://github.com/ShivanshInamdar/cis530

Slides:
https://docs.google.com/presentation/d/1IbZiGizBVxXD841FMwtqKCxyi83rCOg7yyhaqP1wGvg/edit?usp=sharing