The Apple Doesn't Fall Far From the Tree: Reconstructing a Parent Semantic Space from its Descendants

Anonymous EMNLP submission

Abstract

Modern NLP techniques tend to focus on a select few languages with the most resources, while the others are neglected or even ignored. This includes reconstructed proto-languages, which lack a text corpus to train models to represent the semantic meaning and relation of words within it. To this end we have developed a method that uses information from word embedding models from descendant languages to inform a word embedding model for a parent language. We test this method on Latin and two descendants, Spanish and French, through which we are able to recover the information for some of the words in the Latin vocabulary, though not all. We find that the missing words are difficult to recover, though the descendant model still seems to perform well compared to the one that is trained on a Latin corpus.

1 Introduction

With modern NLP techniques, scholars have been able to create a number of powerful tools and resources. However, most NLP research focuses only on a handful of languages (20), ignoring nearly all of the world's 7000 languages spoken today (Magueresse et al., 2020). These languages, categorized as **Low Resource Languages** (LRLs), are "less studied, resource scarce, less computerized, less privileged, less commonly taught, or low density" (Magueresse et al., 2020). Though one typically views LRLs through a modern lens, reconstructed ancestral languages such as Proto-Indo-European (PIE) also fall within this category, though their study is perhaps even more difficult, as the language itself is hypothetical and has no known corpus of texts.

While many scholars have made strides in producing tools for the study of PIE in terms of the etymology of words (UTLRC, 2024), it is currently impossible to model the lexicon of PIE as a semantic network. Progress has been made in the expansion and linking of WordNets for three of the most important Indo-European (IE) languages, Sanskrit, Ancient Greek, and Latin (Zanchi and Ginevra, 2024), though there is currently no way to automatically generate models to help analyze the semantic meaning and relation between words in PIE.

Since we lack a way to create word embedding models without a sizable corpus, and we lack a corpus for reconstructed languages like PIE, we need to get the information from other sources. Taking inspiration from the reconstruction methods used by historical linguists, we propose the use of word embedding models created by descendant languages, including modern languages that have many more resources. Comparing multiple embedding models is normally not possible using word2vec (Mikolov et al., 2013), as the vectors take on arbitrary values between models in the process of training. This can be rectified by aligning models, taking advantage of existing substructures within the models that should remain between languages and making transformations on the word vectors as a whole that maintain these structures while aligning individual vectors (Dev et al., 2021).

In this paper we use pre-aligned word embedding models (Joulin et al., 2018) from two *descendant* languages of Latin to create a new set of vectors for Latin words. We find that this method and some further training on a limited corpus can be used to create a model that is capable of performing well when compared to a model trained without these vectors. We hope that this method can be used to extend the information present in higher resource languages to those with lower resources. Our ultimate aim of this method is to create a word embedding model to help linguists analyze the reconstructed Proto-Indo-European language, as well as to provide a methodology for linguists who study other reconstructed languages and ancient languages with small corpora.

2 Background and Related Works

Word embeddings are a foundational means to process natural language and are used to represent words in a high dimensional vector space, allowing for the capture of a word's features, such as its semantic similarity to other words and its syntactic usage. Normally a large corpus size is required to create such models, like the 100B tokens used in the Google News corpus to train the English word2vec model (Mikolov et al., 2013). These resources are frequently unavailable for lesser-known languages, particularly those lacking written corpora, whether they are modern spoken languages or reconstructed ones such as PIE. Nevertheless, efforts have been made to create word embeddings for LRLs (Coto-Solano, 2022; Fesseha et al., 2021), sometimes taking advantage of information that is present but not used in traditional means to create word embeddings (Jiang et al., 2018). Other times LRLs are augmented with extra information by training on bilingual lexicons (Adams et al., 2017), on parallel languages (Wada et al., 2019), and using a high resource language as an *anchor* in conjunction with a LRL (Eder et al., 2020). Our method differs in that it attempts to take multiple higher resource languages and create a word embedding model for a lower resource parent language.

Our method posits that word embedding models can be aligned – independently trained models can be mapped to a space where the vectors for parallel words are directly comparable to each other. In *Closed Form Word Embedding Alignment*, Dev et al. have created a family of techniques that use simple closed-form transformations on an entire model to align the underlying vectors without disturbing higher level structures. This method was used to align models for two LRLs to a higher resource language to potentially aid in translation though it relies on a consistent set of vocabulary between models. Other works focus on unsupervised alignment (Grave et al., 2019; Alaux et al., 2018) using Wasserstein Procrustes, which is expanded on in *Loss in Translation* by Joulin et al. 2018. For our method we use the aligned models provided in their work.

2.1 Proto-Indo-European Background

Proto-Indo-European (PIE) is a hypothetical language posited by linguists to be the source of all Indo-European (IE) languages, including modern languages such as English, Russian, Welsh, Persian, and Punjabi, as well as ancient languages like Latin, Ancient Greek, and Sanskrit (Renfrew, 1989). Believed to have been spoken around 6000 years ago in the Pontic-Caspian steppe (Bomhard, 2019), PIE was never written down, and thus all of its features must be reconstructed based on indirect evidence found in its descendant languages, seen in grammars, lexicons, and textual evidence.

The primary means of reconstruction is the **Comparative Method** (CM), in which linguists compare words and features that are presumed to derive from a common source, in order to identify what word, grammatical feature, or cultural property was present at an earlier stage (Weiss, 2015). Using this method, linguists have been able to identify how PIE has changed in its descendant languages, including positing regular sound laws that have occurred in the (pre)history of specific languages and language branches. For instance, in the prehistory of English, a certain class of PIE consonants changed, explaining how initial *f-* in English *foot* and *fire* corresponds to *p-* in Ancient Greek *pod-* and *pũr*, respectively. By identifying such changes, linguists are able to reverse-engineer what the original forms would have been, which in this case are PIE **pod-* 'foot' and **páh₂wr̥* '(inanimate) fire'. Taking inspiration from the CM, we are creating a Latin word-embedding model based on its descendants, Spanish and French. Doing so will allow us to validate the proposed method so that we will be able to apply it to languages with no corpora (such as PIE).

3 Methods

In this section, we will first describe our method of preparing existing pre-aligned models of descendant languages and then will describe how these are combined to create the parent model. We use Latin and its descendants, French and Spanish, as a stand in for PIE and its descendants as their models and methods to test them are readily available.

For our descendant vectors we use existing aligned models (Joulin et al., 2018) as a source of vector and word information, specifically the Spanish and French aligned word vectors from Facebook AI Research's fastText (Bojanowski et al., 2017). For each word in each of the model's vocabulary we filter out words that are not within their respective languages. Since these models were trained on French and Spanish Wikipedia articles they include vocabulary that are in a variety of languages,

along with other non-word vocabulary. Because these would require extra time to process and would eventually be ignored in later steps, we opted to remove these. We first filter out words that contain non-language characters, removing around 7.3% and 6.6% of the vocabulary for French and Spanish, respectively. Afterwards we lemmatize the words as a form of stemming them to reduce the amount of work needed for later tasks, further reducing the amount of words in the French and Spanish vocab by 10.6% and 11.9%, respectively. We use spaCy `es_core_news_sm` and `fr_core_news_sm` (Honnibal et al., 2020) for lemmatization. This has the effect of removing words that could not be lemmatized, effectively removing out of language vocab and leaving 82.30% for French and 82.90% Spanish words compared to the original model's vocab.

Next we translate the words into each other's respective languages, which will help us to relate words together later and find which words are common between the two languages. This is done using the Google Translate translation service with the python translation library deep-translator. This same process is done with Latin, but the vocab is taken from the Latin corpus that we use. The text itself comes from the *CLTK* (Johnson et al., 2021) processed version of the Latin *Tesserae Project* (Coffee et al., 2012). These texts are then similarly lemmatized (with LEMLAT 3 (Passarotti et al., 2017)) and any words that cannot be lemmatized due to being out-of-language or containing non-word characters are removed.

Since the Latin vocab is considerably smaller than the French or Spanish vocab and because Google Translate seemed to not always accurately translate the lemmatized versions of the vocab, we translate both the lemmatized and the original words. After gathering the translations between French, Spanish, and Latin we organize these into groups which relate a single word in the original Latin text to a list each of words in French/Spanish that match a translation to or from Latin (from either the lemmatized version of the original word or the original word itself), including any words in French/Spanish that were translated to Spanish/French. If any group has French or Spanish lists that are empty that Latin word is skipped and is considered missing.

These groups are then transformed into their vector representations using their respective models. For each language, French and Spanish, we then calculate the reconstructed Latin vector with the following: find the centroid of the vectors in each language separately to get language-word centers, find the centroid of the two language-word centers to get a inter-language-word center, find the closest vectors in each language to the inter-language-word center by cosine-distance, and finally take the average of these two to get an approximate Latin-word vector.

We found that there were some French/Spanish words that were associated with a Latin word with vectors that were very far from the center, seemingly outliers, and thus used this method to attempt to reduce the influence of the outliers on the final Latin-word vector. We were only able to cover 62.9% of the Latin corpus vocabulary and thus we decided to measure the effects of further training the model using randomly instantiated values for the missing vectors in hopes that the training process would coerce the missing words to take on values that fit with the already established descendant vector information.

4 Experiments and Results

For our experiments we opted to test our descendant Latin model against a model trained without any descendant information, referred to as the normal model. Preliminary tests showed that the effects of using descendant information disappears when training models on a larger corpus, so we opted to limit the size of our training corpus to simulate LRLs with much more limited corpora. We train the models on a corpus size ranging from one-fifth to one-fiftieth of the size of our corpus, the CLTK processed version of the Latin Tesserae Project, which resulted in average ranges of approximately 65000 to 6500 tokens respectively. These were shuffled to remove potential bias, so the exact number of tokens differ.

We also wanted to test the influence of preventing the descendant information from being updated, an experimental lock-factor in the Gensim package. This value ranges from 0 to 1 with a value of 0.0 preventing all updates while a value of 1.0 fully allowing updates. For all non-descendant vectors this value was set to 1 while descendant vectors vary this value.

We train the models using word2vec (Mikolov et al., 2013) using a vector size of 300, an initial learning rate to 0.05, a negative sampling parame-

ter of 10, train on only 20 epochs to prevent overfitting with our small corpus sizes, and leave all other parameters with their default values. Parameters were optimized using Bayesian optimization (Gardner et al., 2014) on a model trained without the descendant language vectors and scored with OddOneOut (Stringham and Izbicki, 2020) as it is designed for use with low-resource languages compared to standard evaluation methods. We found that for most parameters that there is little effect when setting them away from their default aside from the ones mentioned and thus opted to leave their default values.

These parameters were used to train both the descendant Latin models and normal Latin models that we compare against. For each combination of lock-factor and corpus size ratio we run 15 trials. For our normal model we also vary the lock-factor but this parameter is ignored in the normal model so we treat these as extra trials resulting in 90 trials for each corpus size ratio. The results of these tests can be seen in Figure 1 which shows both the results of the varying the lock-factor in the descendant model and the normal model.

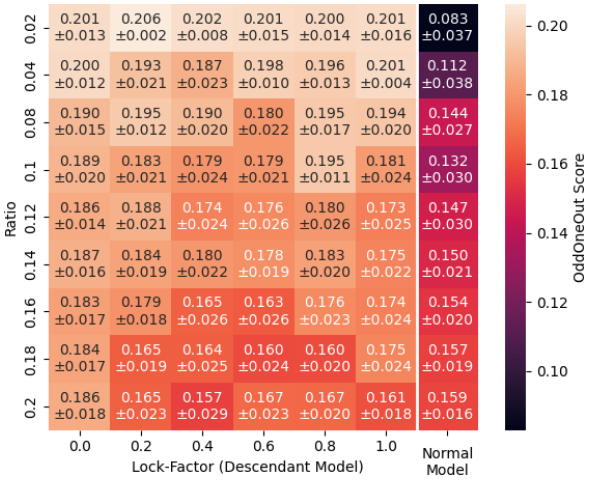

Figure 1: Mean and Std. scores for values of corpus size (ratio) and lock-factor for both the Descendant models (left) and Normal models (right)

5 Discussion

Our results indicate that the process works with mixed success. We are able to create a model using descendant information that is capable of performing reasonably on Latin tests with OddOneOut despite none of the descendant information directly using any Latin vocabulary. This by itself suggests that this method has merit, though it is imperfect.

Primarily we find that the performance of our descendant models tends to decrease as the corpus size increases. This might be due to how updates are limited when using the lock-factor. We do not have a detailed understanding of how this parameter affects the training process as it is not well documented and is considered an experimental feature. This also may be due to some information being unlearned. As the model is being trained the values of the unlocked vectors may take on values that are appropriate to the corpus, but are misaligned from the locked vectors. In the future we plan on finding more robust and understood ways to limit updating the values for vectors that we consider good.

For smaller corpora sizes our method tends to outperform the normal model and continues to do so even up to one-fifth the size of the corpus. After this the normal model overtakes the descendant model and at higher lock-factors, allowing the vectors to be updated leads to much lower scores. Allowing for the full corpus and a lock-factor of 1 gives an OddOneOut score of 0.13 in the descendant model compared to a 0.20 for the normal model. This further suggests that the process of training on the corpus interferes with the existing vectors.

6 Conclusion

Our initial process of using Latin descendant languages to create Latin vectors shows promise, as it can perform similarly to a model trained without this information despite not using any Latin information itself, and not generating values for all words in the vocab. Filling in the missing vocab with suitable vectors remains difficult to do and limiting the ability for the model to update these vectors does not seem to have the intended effect. Training word embeddings on languages that are missing suitable corpora, such as Proto-Indo-European, is not possible, so this method of recovering some information using descendant languages is still suitable in these cases. In the future we plan on both reducing the need to recover the missing vocab by improving our method of generating the vectors and also improving the method by which vectors for the missing vocab are created.

7 Limitations

In developing our method we found limitations in both the tools we used and in developing the method itself. Our main limitation is in how we measure the efficacy of the model. Standard tests like analogy sets are difficult to obtain for LRLs, and their translations may not be valid for older languages whose usage and concepts are not transferable to modern languages (Izbicki, 2022). Since we are mimicking a language with even lower resources, we are limited in how we can measure the model as the lack of tokens in the corpus leads to performance that is difficult to measure using standard means. We opted to make use of the OddOneOut method introduced by Stringham and Izbicki 2020 as it directly addresses those limitations, but the low performance of the other method introduced in this work, Topk, meant that we could not use it. This does not seem to be a limitation solely due to corpus size as Topk performs worse even when trained on the entire Latin corpus. Part of this could be the difficulty in selecting appropriate categories outside of those that were listed in their documentation for the library[1], as many of the categories we attempted to use came up empty despite seeming to be well suited for the Latin language, and thus we opted to use those that we knew worked well. In the future we plan on expanding how we use these evaluation metrics to get a better understanding of the performance of our model.

We also have limitations in how our models were created. While many methods for model alignment were explored, we had difficulties in training our own good quality models and had even more difficulties when attempting to replicate any of the alignment methods we found. Part of this is due to the lack of existing code that implements these methods, with any existing code that we found often being poorly documented and relying on specific structures that would need to be reverse-engineered from the code. This process was too time consuming for us and the availability of pre-aligned vectors meant that we moved forward with these pre-aligned vectors. These models do not exist for the majority of the languages that we wish to apply this process to as we are looking to ultimately create word vectors for PIE. In the future we plan on using our own aligned models.

Additionally our process is making use of various inexact methods for the sake of convenience. Ideally the translation process would use a more trusted method of finding which words are related between languages, one that would take into account how words change as they move from language to language. We initially planned on making use of Wiktionary as it does contain such information, but it is occasionally incomplete and is still largely meant to be mostly human-readable and not easily machine processable. There are efforts to make use of this data (Izbicki, 2022) that we plan on incorporating with our own method. We also only make use of two Romance languages, French and Spanish, as it allowed us to save time, and we plan on incorporating additional Romance languages in future projects. We also recognize that the Latin language used in its texts is a literary register, one which should not be viewed as an exact parent of the Romance languages, rather an approximate one.

[1] https://github.com/n8stringham/gensim-evaluations

References

Oliver Adams, Adam Makarucha, Graham Neubig, Steven Bird, and Trevor Cohn. 2017. Cross-lingual word embeddings for low-resource language modeling. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 937–947.

Jean Alaux, Edouard Grave, Marco Cuturi, and Armand Joulin. 2018. Unsupervised hyperalignment for multilingual word embeddings. *arXiv preprint arXiv:1811.01124*.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Allan R Bomhard. 2019. The Origins of Proto-Indo-European: The Caucasian Substrate Hypothesis. *Journal of Indo-European Studies*, 47.

Neil Coffee, Jean-Pierre Koenig, Shakthi Poornima, Christopher W Forstall, Roelant Ossewaarde, and Sarah L Jacobson. 2012. The tesserae project: intertextual analysis of latin poetry. *Literary and linguistic computing*, 28(2):221–228.

Rolando Coto-Solano. 2022. Evaluating word embeddings in extremely under-resourced languages: A case study in bribri. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4455–4467.

Sunipa Dev, Safia Hassan, and Jeff M Phillips. 2021. Closed form word embedding alignment. *Knowledge and Information Systems*, 63(3):565–588.

Tobias Eder, Viktor Hangya, and Alexander Fraser. 2020. Anchor-based bilingual word embeddings for low-resource languages. *arXiv preprint arXiv:2010.12627*.

Awet Fesseha, Shengwu Xiong, Eshete Derb Emiru, Moussa Diallo, and Abdelghani Dahou. 2021. Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. *Information*, 12(2):52.

Jacob R Gardner, Matt J Kusner, Zhixiang Eddie Xu, Kilian Q Weinberger, and John P Cunningham. 2014. Bayesian optimization with inequality constraints. In *ICML*, volume 2014, pages 937–945.

Edouard Grave, Armand Joulin, and Quentin Berthet. 2019. Unsupervised alignment of embeddings with wasserstein procrustes. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1880–1890. PMLR.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. spaCy: Industrial-strength Natural Language Processing in Python.

Mike Izbicki. 2022. Aligning word vectors on low-resource languages with wiktionary. In *Proceedings of the Fifth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2022)*, pages 107–117.

Chao Jiang, Hsiang-Fu Yu, Cho-Jui Hsieh, and Kai-Wei Chang. 2018. Learningword embeddings for low-resource languages by pu learning. *arXiv preprint arXiv:1805.03366*.

Kyle P Johnson, Patrick J Burns, John Stewart, Todd Cook, Clément Besnier, and William JB Mattingly. 2021. The classical language toolkit: An nlp framework for pre-modern languages. In *Proceedings of the 59th annual meeting of the association for computational linguistics and the 11th international joint conference on natural language processing: System demonstrations*, pages 20–29.

Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. 2018. Loss in translation: Learning bilingual word mapping with a retrieval criterion. *arXiv preprint arXiv:1804.07745*.

Alexandre Magueresse, Vincent Carles, and Evan Heetderks. 2020. Low-resource languages: A review of past work and future challenges. *arXiv preprint arXiv:2006.07264*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Marco Passarotti, Marco Budassi, Eleonora Litta, and Paolo Ruffolo. 2017. The lemlat 3.0 package for morphological analysis of latin. In *Proceedings of the NoDaLiDa 2017 workshop on processing historical language*, pages 24–31.

Colin Renfrew. 1989. The origins of indo-european languages. *Scientific American*, 261(4):106–115.

Nathan Stringham and Mike Izbicki. 2020. Evaluating word embeddings on low-resource languages. In *Proceedings of the First Workshop on Evaluation and Comparison of NLP Systems*, pages 176–186.

UTLRC. 2024. The Linguistics Research Center. Indo-European Lexicon (IELEX): Pie Etyma and IE Reflexes.

Takashi Wada, Tomoharu Iwata, and Yuji Matsumoto. 2019. Unsupervised multilingual word embedding with limited resources using neural language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3113–3124.

Michael Weiss. 2015. The Comparative Method. In *The Routledge Handbook of Historical Linguistics*, pages 127–145. Routledge.

Chiara Zanchi and Riccardo Ginevra. 2024. Linked Wordnets for Ancient Indo-European Languages.