# Orthographic vs. Semantic Representations for Unsupervised Morphological Paradigm Clustering

**E. Margaret Perkoff**[⋆] **Josh Daniels**[†] **Alexis Palmer**[†]
[⋆]Dept. of Computer Science, [†]Dept. of Linguistics
University of Colorado Boulder
{margaret.perkoff, joda4370, alexis.palmer}@colorado.edu

## Abstract

This paper presents two different systems for unsupervised clustering of morphological paradigms, in the context of the SIGMORPHON 2021 Shared Task 2. The goal of this task is to correctly cluster words in a given language by their inflectional paradigm, without any previous knowledge of the language and without supervision from labeled data of any sort. The words in a single morphological paradigm are different inflectional variants of an underlying lemma, meaning that the words share a common core meaning. They also - usually - show a high degree of orthographical similarity. Following these intuitions, we investigate KMeans clustering using two different types of word representations: one focusing on orthographical similarity and the other focusing on semantic similarity. Additionally, we discuss the merits of randomly initialized centroids versus pre-defined centroids for clustering. Pre-defined centroids are identified based on either a standard longest common substring algorithm or a connected graph method built off of longest common substring. For all development languages, the character-based embeddings perform similarly to the baseline, and the semantic embeddings perform well below the baseline. Analysis of the systems' errors suggests that clustering based on orthographic representations is suitable for a wide range of morphological mechanisms, particularly as part of a larger system.

## 1 Introduction

One significant barrier to progress in morphological analysis is the lack of available data for most of the world's languages. As a result, there is a dramatic divide between high and low resource languages when it comes to performance on automated morphological analysis (as well as many other language-related tasks). Even for languages with resources suitable for computational morphological analysis, there is no guarantee that the available data in fact covers all important aspects of the language, leading to significant error rates on unseen data. This uncertainty regarding training data makes unsupervised learning a natural modeling choice for the field of computational morphology. The unsupervised setting takes away the need for large quantities of labeled text in order to detect linguistic phenomena. The SIGMORPHON 2021 shared task aims to leverage the unsupervised setting in order to identify morphological paradigms, at the same time including languages with a wide range of morphological properties.

| Surface Forms | | Morphological Features |
|---|---|---|
| walk | bring | V; 1SG; 2SG; 3PL; 1PL |
| walks | brings | V; 3SG |
| walking | bringing | PRES; PART |
| walked | brought | PAST |

Table 1: Morphological paradigms for the English verbs *walk* and *bring*.

For a given language, the morphological paradigms are the models that relate root forms (or lemmas) of words to their surface forms. The task we tackle is to cluster surface word forms into groups that reflect the application of a morphological paradigm to a single lemma. The lemma of the paradigm is typically the dictionary citation form, and the corresponding surface forms are inflected variations of that lemma, conveying grammatical properties such as tense, gender, or plurality. For example, Table 1 displays partial clusters for two English verbs: *walk* and *bring*.

In developing our system, we consider two types of information that could reasonably play a role in unsupervised paradigm induction. First, the words in a single paradigm cluster are different inflectional variants of an underlying lemma, meaning



*Proceedings of the Seventeenth SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*,pages 90–97

that the words share a common core meaning. They also - usually - show a high degree of orthographical similarity. Following these intuitions, we investigate KMeans clustering using two different types of word representations: one focusing on orthographical similarity and the other focusing on semantic similarity. Intuitively, we would expect the cluster of forms for *walk* to be recognizable largely based on orthographic similarity. The partially irregular cluster for *bring* shows greater orthographical variability in the past-tense form *brought* and so might be expected to require information beyond orthographic similarity.

**System Overview.** The core of our approach is to cluster unlabelled surface word forms using KMeans clustering; a complete architecture diagram can be seen in Figure 1. After reading the input file for a particular language to identify the lexicon and alphabet, we transform each word into two different types of vector representations. To capture semantic information, we train Word2Vec embeddings from the input data. The orthography-based representations we learn are character embeddings, again trained from the input data. Details for both representations appear in section 4.1. For the experiments in this paper, we test each type of representation separately, using randomly initialized centers for the clustering. In later work, we plan to explore the integration of both types of representations. We would also like to explore the use of pre-defined centers for clustering. These pre-defined centers could be provided using either a longest common subsequence method or a graph-based algorithm such as that described in section 4.3. The final output of the system is a set of clusters, each one representing a morphological paradigm.

## 2 Previous Work

The SIGMORPHON 2020 shared task set included an open problem calling for unsupervised systems to complete morphological paradigms. For the 2020 task, participants were provided with the set of lemmas available for each language (Kann, 2020). In contrast, the 2021 SIGMORPHON task 2 outlines that submissions are unsupervised systems that cluster input tokens into the appropriate morphological paradigm (Nicolai et al., 2020). Given the novelty of the task, there is a lack of previous work done to cluster morphological paradigms in an unsupervised manner. However, we have identified key methods from previous work in computational morphology and unsupervised learning that could be combined to approach this problem.

Previous work has identified the benefit of combining rules based on linguistic characteristics with machine learning techniques. Erdmann et al. (2020) established a baseline for the Paradigm Discovery Problem that clusters the unannotated sentences first by a combination of string similarity and lexical semantics and then uses this clustering as input for a neural transducer. Erdmann and Habash (2018) investigated the benefits of different similarity models as they apply to Arabic dialects. Their findings demonstrated that Word2Vec embeddings significantly underperformed in comparison to the Levenshtein distance baseline. The highest performing representation was a combination of FastText and a de-lexicalized morphological analyzer. The FastText embeddings (Bojanowski et al., 2016) have the benefit of including sub-word information by representing words as character n-grams. The de-lexicalized analyzer relies on linguistic expert knowledge of Arabic to identify the morphological closeness of two words. In the context of the paper, it is used to prune out word relations that do not conform to Arabic morphological rules. The approach mentioned greatly benefits from the use of a morphological analyzer, something that is not readily available for low-resource languages. Soricut and Och (2015) focused on the use of morphological transformations as the basis for word representations. Their representation can be quite accurate for affix-based morphology.

Our representations are based entirely off of unlabelled data and do not require linguistic experts to provide morphological transformation rules for the language. Additionally, we hoped to create a system that would be robust for languages that include non-affix based morphology. In this work we compare Word2Vec representations to character-based representations to represent orthography. We have not yet evaluated additional representations or combinations of the two.

## 3 Task overview

The 2021 SIGMORPHON Shared Task 2 created a call for unsupervised systems that would create morphological paradigm clusters. This was intended to build upon the shared task from 2020 that focused on morphological paradigm completion. Participants were provided with tokenized Bible data from the JHU bible corpus (McCarthy

et al., 2020) and gold standard paradigm clusters for five development languages: Maltese, Persian, Portuguese, Russian and Swedish. Teams could use this data to train their systems and evaluate against the gold standard files as well as a baseline. The baseline provided groups together words that share a substring of length *n* and then removes any duplicate clusters. The resulting systems were then used to cluster tokenized data from a set of test languages including: Basque, Bulgarian, English, Finnish, German, Kannada, Navajo, Spanish, and Turkish.

## 4 System Architecture

The overall architecture of our system includes several distinct pieces as demonstrated in Figure 1. For a given language, we read the corpus text provided and generate a lexicon of unique words. The lexicon is then fed to an embedding layer and an optional lemma identification layer. The embedding layer generates a vector representation of each word based on either a character level embedding or a Word2Vec embedding. When used, the lemma identification layer generates a set of predefined lemmas from the lexicon based on either the standard longest common substring or a connected graph formed from the longest common substring. Result word embeddings along with the optional set of predefined lemmas are used as input to a KMeans clustering algorithm. In the event predefined lemmas are not provided, the system defaults to using a randomly initialized set of centroids. Otherwise, the initial centroids for the clusters are the result of finding the appropriate word embedding for the lemmas identified. Once a cluster has been created, the output cluster predictions are formatted into a paradigm dictionary which can be written to a file for evaluation.

### 4.1 Word Representations

We create two different types of word representations, aiming to capture information that may reflect the relatedness of words within a paradigm.

**Character Based Embeddings.** To capture orthographic information, we generate a character-based word embedding for the language. For each language we do the following:

1. Generate a lexicon of all the words in the development corpus and an alphabet of unique characters in the language.
2. Identify the maximum word length of the lexicon.
3. Create a dictionary of the alphabet where each character corresponds to a float value between 0 (non-inclusive) and 1 (inclusive).
4. For each word:
   (a) Initialize an array of zeros the same size as the maximum length word.
   (b) Map each character in the word, in order, to its respective float value based on our alphabet dictionary. Leave the remaining values as zero.

This representation focuses purely on the characters of the language. For the time being, it does not take into account the relationship between orthographic characters in any of the languages but future work could attempt to create smarter numerical representations based on these relationships.

**Word Embeddings with Word2Vec.** To incorporate semantic and syntactic information, we use the Word2Vec embeddings. Specifically, we train a Word2Vec model for each language with the Gensim skip-gram representations (Řehůřek and Sojka, 2010).

### 4.2 (Optional) Lemma Identification

**LCS Graph Formation** One of the challenges of using clustering-based methods on this problem is determining the number of morphological paradigms expected to be present and then finding suitable lemmas for each to serve as centers for clustering. One potential approach to find lemmas is to first arrange the words into a network graph based on the longest common substring relationships between them. Specifically, for each attested word W in a language's data, the longest common substring (LCS) is calculated between W and every other attested word in the language. Graph edges are then constructed between W and the word (or words if there are multiple with the same length LCS) that have the longest LCS with W. This process is repeated for every word in the given language's corpus. This results in a large graph that appears to capture many of the morphological dependencies within the language.

Next, we split the graph into highly connected subgraphs (HCSs). HCS are defined as graphs in which the number of edges that would need to be cut to split the graph into two disconnected

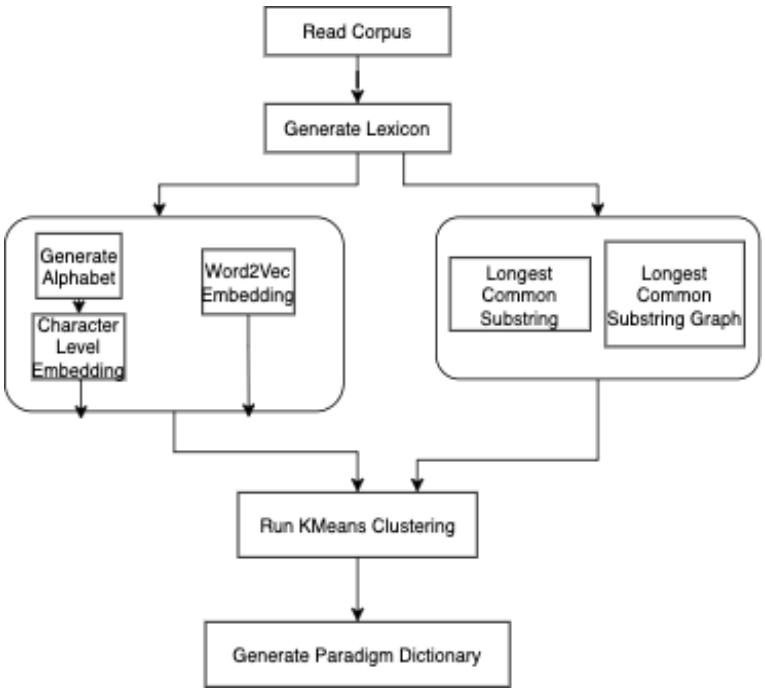


Figure 1: Overall Statistical Clustering Architecture diagram. There are two possible word embedding algorithms represented in the diagram (left side of split). The optional lemma identification layer also includes two possible methods (right side).

subgraphs is greater than one half of the number of nodes. This is helpful because in the LCS graphs generated, morphologically related forms tend to be connected relatively densely to each other and only weakly connect to forms from other paradigms. Additionally, the use of a threshold based algorithm like HCS, unlike other clustering methods, would allow lemmas to be extracted without having to prespecify the expected number of lemmas beforehand. Unfortunately, during testing the HCS graph analysis proved computationally taxing and was unable to be completed in time for evaluation, though qualitative analysis of the generated LCS graphs suggests the technique may still be useful with better computational power. We will explore this method further in future work.

### 4.3 Clustering

The word representations described in section 4.1 are used as input to a clustering algorithm. We use the KMeans algorithm as defined by the sklearn implementation (Pedregosa et al., 2011). The KMeans approach is one of the pioneering algorithms in unsupervised learning (MacQueen et al., 1967). Input values are grouped by continuously shifting clusters and their centers while attempting to minimize the variance of each cluster. This indicates that the cluster that a particular word is assigned to should be as close (as defined by Euclidean distance) to the cluster's center, or the lemma word, as possible.

**Clustering with Randomly Initiated Centers.** For comparison, we evaluate the effectiveness of using randomly initialized centers for our clusters. In the context of this task, this means that the first set of centers fed to the algorithm do not necessarily correspond to any valid word in the given language, or perhaps any language. Another obstacle for this approach in an unsupervised setting is defining the number of clusters to use. Identifying this requires human interference with hyper-parameters that are not going to be cross-linguistically relevant. The size of the input bible corpus and the inflectional morphology of the language both directly impact the number of clusters, or the number of lemmas, that are relevant. We used a range of cluster sizes for the development languages from 100 to 6000 to evaluate which ones provided the highest accuracy. For the test languages, we chose to submit results for clusters of size 500, 1000, 1500, and 1900 to assess performance variability based on number of lemmas.

**Extension: Initializing with Non-Random Centers.** The use of non-random centers would have multiple benefits in the context of this task. This approach would incorporate linguistic information

| Language | BL | KMW2V | KMCE |
|---|---|---|---|
| Maltese | **0.29** | 0.19 | 0.25 |
| Persian | 0.30 | 0.18 | **0.36** |
| Portuguese | **0.34** | 0.06 | 0.24 |
| Russian | **0.36** | 0.11 | 0.34 |
| Swedish | 0.44 | 0.18 | **0.45** |

Table 2: F1 Scores for each of the model types on all development languages. The best F1 scores are in bold. BL is Baseline, KMW2V is KMeans with Word2Vec embeddings, and KMCE is KMeans with Character Embeddings.

| Language | BL | 500 | 1000 | 1500 | 1900 |
|---|---|---|---|---|---|
| Basque | 0.21 | 0.29 | **0.31** | 0.27 | 0.29 |
| Bulgarian | **0.39** | 0.21 | 0.27 | 0.29 | 0.30 |
| English | **0.52** | 0.29 | 0.37 | 0.43 | 0.45 |
| Finnish | **0.29** | 0.19 | 0.24 | 0.26 | 0.27 |
| German | 0.38 | 0.26 | 0.33 | 0.38 | **0.40** |
| Kannada | 0.24 | 0.29 | **0.30** | **0.30** | 0.29 |
| Navajo | 0.33 | 0.33 | 0.38 | 0.39 | **0.41** |
| Spanish | **0.39** | 0.24 | 0.29 | 0.30 | 0.31 |
| Turkish | **0.25** | 0.16 | 0.20 | 0.22 | 0.22 |

Table 3: F1 Scores for the baseline (BL) and the KMCE models on the test languages. The best F1 scores are in bold. Test languages were evaluated on KMCE models with clusters of size 500, 1000, 1500, and 1900.

to inform the initial set of centers. This could lead to quicker convergence of a model due to more intelligently picked centers. It could also prevent the model from being skewed towards less than ideal center values. Additionally, with pre-defined centers we can remove the need to arbitrarily define the number of clusters.

In the scope of this task, we were unable to experiment with pre-defined center values but we have proposed two potential methods for doing so: using longest common substrings and picking highly connected nodes from an LCS graph formation. The longest common substring approach would mimic the lemma identification approach described above (4.2). Both of these systems are represented as an optional lemma identification layer on the right hand side of Figure 1. The output of each one would be a set of words to use as centers. Each word would be converted to the appropriate word representation and then fed as an input to the KMeans clustering.

## 5 Results

Table 2 shows results to date. We compare the two representation methods on the development languages. The KMeans clusterings for the development languages were generated based on optimal cluster values starting with size 100 and increasing to a cluster size of 6000, or until the accuracy no longer improved from an increase in cluster size. For the Word2Vec embeddings we used clusterings of size 110 for Maltese, 130 for Persian, 1490 for Portuguese, 1490 for Russian, and 1490 for Swedish. With the character embeddings, we had 540 clusters for Maltese, 110 clusters for Persian, 2200 clusters for Portuguese, 4000 clusters for Russian, and 5400 clusters for Swedish. The F1 scores provided are based on comparing the appropriate model's predictions to the gold paradigms for this task using the evaluation function defined in the SIGMORPHON 2021 Task 2 github repository. The KMCE models clearly and consistently outperform the KMW2V models, for all development languages.

For test languages, we run clustering only with the better-performing character-based representations. The performance on test languages was evaluated with clusters of size 500, 1000, 1500, and 1900. These results are in Table 3. We found that our algorithm outperformed the baseline for Basque, German, Kannada, and Navajo. For both Basque and Kannada, the largest clustering did not have the highest result suggesting that the corpora provided for these languages contain a smaller number of morphological paradigms. In the case of Bulgarian, English, Spanish, and Finnish, we note that the KMCE model performance increases with each increase in cluster size. This suggests that the model accuracy would continue increasing if we ran the model for these languages with a higher number of clusters. Additional discussion of the error analysis appears in section 6, and fo the results in section 7.

## 6 Error Analysis

We have evaluated the results from the Word2Vec representations and our character-based embedding and compared them to the gold standard paradigms provided by the task organizers. We have found that, overall, the character-based version is more robust on regular verb forms than the Word2Vec version, and that neither is effective on irregular forms. Additionally, we explore some of the nu-

anced errors with the character based embeddings and how they could be addressed for future work.

### 6.1 Regular Verb Forms

Our results are consistent with our initial expectation that an orthographic word representation would perform better on regular verb forms than the Word2Vec representation, since it weights closeness based on the characters of the word. The character embedding correctly groups many surface forms together based on regular English morphological paradigms, or those that follow the pattern of *-ed* for past tense, *-s* for third person singular present, - ∅ for first, second and third person plural present. However, there were sometimes words missing from the paradigm. For example, the system generated the paradigm {*stumble, stumbles, stumbling*}. This should have included *stumbled*, but instead that is in a paradigm with *thaddaeus*. In contrast, the Word2Vec representation separates all four surface forms into different morphological paradigms. For Spanish paradigms, we see that the character embeddings perform well for matching some of the regular surface forms together, but cannot handle longer suffixes. For example, *aprendas, aprendan* and *aprenden* are grouped together while leaving out longer surface forms like *aprenderemos*. Similarly, *hablaste, hablaras, hablaran* and *hablara* are grouped in the same morphological paradigm, but *hablar, hablan, hablar, hables* and *hablen* are part of a separate grouping. We discuss the issue of errors related to word length in detail below.

### 6.2 Irregular Verb Forms

Because it focuses on semantic relatedness, we expected the Word2Vec representation to be more accurate in grouping together irregular surface forms from the same paradigm. For example, we have the paradigm for *go*: {*go, goes, going, went*}. In fact, Word2Vec created a morphological paradigm for *go*, one for {*upstairs, carry, goes, favour*}, {*going, reply, robbed, dared, gaius, failed, godless*}, and one for *went*. The orthographical representation also produced some undesired results, with a paradigm for { *eyes, goat, goes, gone, gong, else, eloi, noah, none, sons, long*} and {*gains, noisy, lasea, lysias, gates, lying, fatal, often, notes, loses, latin, latest, going*}. The other surface forms of *went* and *go* also ended up in separate morphological paradigms. These results suggest that neither representation is currently robust enough to handle irregular verb forms.

### 6.3 Character Distance Errors

In some cases, the character representations result in strange cluster formations due to the usage of Euclidean distance in the sklearn KMeans library. Since each character in the language's alphabet was mapped arbitrarily to a numeric value, the closeness of a pair of characters does not reflect a morphological relationship between those symbols. However, characters that are assigned to numerical values that are closer to one another will be classified as closer by the Euclidean distance algorithm. It would be possible to learn more about the language specific character relations by training a recurrent network with a focus on the character sequence alignments. This network could then be used as an encoder to generate character level embeddings.

### 6.4 Non-Affix Based Morphology

For verb forms in English that do not use a regular affixation paradigm, we find that some surface forms are paired together in the correct clusters, but those clusters often contain additional unrelated words. Consider the following group: {*drank, drinks, drink, drunk, breaks, break, branch*}. In this cluster, we see that *drank, drinks, drink* and *drunk* were all correctly identified as being related. The algorithm also matched *break* and *breaks* together. This suggests that character representations have the potential to identify morphological transformations that occur at different points in the word, as opposed to just prefixes or suffixes. However, the result is a combination of what should be three distinct morphological paradigms, including a unique paradigm for *branch*. In Navajo, *jidooleeł* and *dadooleeł* are correctly put in the same paradigm. However, this paradigm also includes *jidooleełgo* and *dadooleełgo*, which are not morphologically related. We also see the tendency of over-grouping in Basque, where *bitzate, nitzan*, and *ditzan* are all grouped together along with over ten other unrelated forms. This could potentially be addressed by increasing the number of clusters to favor smaller clusters. Adding semantic features to the word embeddings such as part of speech or limited context windows may also help filter out words that are not relevant to a particular paradigm.

### 6.5 Word Length

Another type of cluster error has to do with word length. The word representation vectors were sized based on the largest word present in a given language's corpus. If a word is under the maximum length, the remaining vector gets filled in with zeros. This means that words that are similar in length are more likely to be paired together for a cluster. The gold data created the morphological paradigm {*crowd, crowds, crowding*}, while ours created two separate clusterings: {*crowd, crowds*} and {*brawling, proposal, crowding*}. This is also present in the clustering of certain words in Navajo. Our algorithm grouped *nizaad*, and *bizaad* together, but some of the longer forms in this paradigm were excluded such as *danihizaad* and *nihizaad*. In future work, we would attempt to mitigate this by using subword distances or cosine similarity as the basis for distance metrics in a clustering algorithm. This could prevent inaccurate groupings due to large affix lengths.

## 7 Discussion, Conclusions, Future Work

Overall, these results demonstrate an improvement over the baseline in several languages, namely Persian, Swedish, Basque, Germany, Kannada, and Navajo, when using KMeans clustering over character embeddings. This suggests that embedding-based clustering systems merit further exploration as a potential approach to unsupervised problems in morphology. The fact that the character embedding system outperformed the W2V one and the fact that performance was strongest on words with regular inflectional paradigms suggests that this approach might be best suited to synthetic and agglutinating languages in which morphology is encoded fairly simply within the orthography of the word. Languages that rely heavily on more complex morphological processes, particularly non-concatenative morphology, would likely require an extension of this system that integrates more sources of non-orthographic information, or a different approach all together.

One obvious avenue for building on this research is to find more efficient and more effective methods for the initial process of lemma identification. Developing a set of lemmas would allow a pre-defined set of centers to be fed into the clustering algorithm rather than using randomly defined centers, which would likely improve performance. This could be done by leveraging an initial rule based analysis or through the threshold-based graph clustering technique discussed above. Other potential variations on that approach, once the problem of computational limits has been solved, include using longest common sequences rather than longest common substrings, and weighting graph edges by the length of the LCS between the two words. The former would potentially help accommodate forms of non-concatenative morphology, while the latter would potentially include more information about morphological relationships than an unweighted graph does. Future research should also explore how other sources of linguistic information could be leveraged for this task. This could include other forms of semantic information outside of the context-based semantics used by W2V, as well as things like the orthographic-phonetic correspondences in a given language.

Finally, we would like to explore filtering of the output clusters according to language-specific properties in order to improve the overall results.This would involve adding additional layers to our system architecture that take place after a distance-based clustering. One such layer could prune unlikely clusters based off of a morphological transformations, such as the method used by Soricut and Och (2015). Future unsupervised systems for clustering morphological paradigms should consider the benefits of hierarchical models that leverage different algorithm types to gain the most information possible.

## References


Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information.

Alexander Erdmann, Micha Elsner, Shijie Wu, Ryan Cotterell, and Nizar Habash. 2020. The paradigm discovery problem. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7778–7790, Online. Association for Computational Linguistics.

Alexander Erdmann and Nizar Habash. 2018. Complementary strategies for low resourced morphological modeling. In *Proceedings of the Fifteenth Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 54–65, Brussels, Belgium. Association for Computational Linguistics.

Katharina Kann. 2020. Acquisition of inflectional morphology in artificial neural networks with prior knowledge. In *Proceedings of the Society for Computation in Linguistics 2020*, pages 144–154, New

York, New York. Association for Computational Linguistics.

MacQueen, J, and author. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, pages 281–297. University of California Press.

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Garrett Nicolai, Kyle Gorman, and Ryan Cotterell, editors. 2020. *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*. Association for Computational Linguistics, Online.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Radu Soricut and Franz Och. 2015. Unsupervised morphology induction using word embeddings. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1627–1637, Denver, Colorado. Association for Computational Linguistics.

Radim Řehůřek and Petr Sojka. 2010. Software framework for topic modelling with large corpora. In *Proceedings of LREC 2010 workshop New Challenges for NLP Frameworks*, pages 46–50, Valletta, Malta. University of Malta.