# Multilanguage Word Embeddings for Social Scientists: Estimation, Inference, and Validation Resources for 157 Languages[*]

Elisa M. Wirsching [†]
Pedro L. Rodriguez [‡]
Arthur Spirling [§]
Brandon M. Stewart [¶]

**Abstract**

Word embeddings are now a vital resource for social science research. However, obtaining high-quality training data for non-English languages can be difficult, and fitting embeddings therein may be computationally expensive. In addition, social scientists typically want to make statistical comparisons and do hypothesis tests on embeddings, yet this is non-trivial with current approaches. We provide three new data resources designed to ameliorate the union of these issues: (1) a new version of `fastText` model embeddings; (2) a multi-language "a la carte" (ALC) embedding version of the `fastText` model; (3) a multi-language ALC embedding version of the well-known `GloVe` model. All three are fit to Wikipedia corpora. These materials are aimed at "low resource" settings where the analysts lack access to large corpora in their language of interest or to the computational resources required to produce high-quality vector representations. We make these resources available for 40 languages, along with a code pipeline for another 117 languages available from Wikipedia corpora. We extensively validate the materials via reconstruction tests and other proofs-of-concept. We also conduct human crowdworker tests for our embeddings for Arabic, French, (traditional Mandarin) Chinese, Japanese, Korean, Russian, and Spanish. Finally, we offer some advice to practitioners using our resources.

[*] The resources discussed in this paper can be found here: http://alcembeddings.org/

[†] PhD candidate, Wilf Family Department of Politics, New York University (elisa.wirsching@nyu.edu)

[‡] Visiting Scholar, Center for Data Science, New York University, United States; and International Faculty, Instituto de Estudios Superiores de Administración, Venezuela, (pedro.rodriguez@nyu.edu)

[§] Professor of Politics, Princeton University (as1780@princeton.edu)

[¶] Associate Professor, Sociology and Office of Population Research, Princeton University (bms4@princeton.edu)

# 1 Motivation

Word embeddings (e.g. Mikolov et al., 2013) are now an important tool of social science. In contrast to traditional ways of representing the contents of documents, these estimated real-valued vectors enable us to talk more directly about the 'meanings' and connotations of terms in natural language (Caliskan, Bryson and Narayanan, 2017; Rodman, 2020). Applications include modeling political emotions (e.g. Gennaro and Ash, 2022) and legislative ideology (e.g. Rheault and Cochrane, 2020). At least two challenges remain: First, obtaining high-quality embeddings for non-English languages can be difficult. Second, it has proved non-trivial to place embeddings in a modeling framework, such that one can answer questions of the form "does this group differ in a statistically significant way in terms of their embeddings of a given term"? Here, we provide resources for the union of these issues. We use the embedding models and multilingual data from the `fastText` project of Grave et al. (2018) and combine it with recent advances in "a la carte" (ALC) embeddings (Khodak et al., 2018). The latter can then be seamlessly placed in a regression-style setup courtesy of Rodriguez, Spirling and Stewart (2023).

## 1.1 New `fastText` Embeddings

The `fastText` project underpins the first contribution and provides two types of resources: first, an (open source) modeling architecture "that allows users to learn text representations"[1]. Second, the output of applying that embedding model to 157 languages for which training data comes from *Common Crawl* and Wikipedia. A strength of the `fastText` model is that it uses *subword* information in addition to the usual context word arrangement for prediction. This can result in higher quality embeddings than for whole words (only) because tokens that are not identical but that contain similar parts (like `policy` and `policies`) are not treated as completely separate entities. This is helpful when, say, a specific form of a word was rare in the training documents but for which we still have some information from other tokens that were more common.[2]

On inspection, we saw that *Common Crawl* includes many typos and rare terms (plus many

[1] As described here: https://fasttext.cc/

[2] See Supporting Information (SI) A for more information.

English loan words). Beyond this potential for noise, *Common Crawl* is not separated by language—it is one combined corpus that requires non-trivial division for the end-user we have in mind here. Our first contribution is simply taking the `fastText` *pipeline* and fitting it to Wikipedia in various languages. Thus, we have "our" version of `fastText`, which is cleaner than the original (though the training domain is admittedly more restricted).[3]

## 1.2 New ALC Embeddings and Transformation Matrices

Our second set of contributions is to produce ALC embeddings. First, for this "new" version of `fastText`. Second, we provide ALC embeddings for `GloVe`, which we also trained on Wikipedia corpora. Details on these embeddings can be found in the SI[4] but the logic is straightforward. Essentially, embeddings of a given word $w_v$ are estimated by taking the mean of the pre-trained embeddings of the tokens around it ($u_w$) and then using a *transformation matrix* (denoted $\mathbf{A}$) to redirect the new embedding away from common directions in the embeddings space (e.g., function words) otherwise likely to be over-represented in that averaging process. This allows analysts to produce high-quality vector representations even when they have very little data—including *single* instances of terms, assuming one has the context of that word and a sufficiently large corpus to pre-train embeddings. This, in turn, facilitates *statistical* inference because one can place the embeddings on "the left-hand side" and covariates of interest as predictors: for this purpose, Rodriguez, Spirling and Stewart (2023) give machinery for estimating both coefficients (on, say, group membership variables) and uncertainty around them. We provide those required (reasonable) pre-trained embeddings using both `fastText` and `GloVe` models applied to Wikipedia and the relevant learned transformation matrix. We note that while there certainly are other non-English language embedding resources (e.g. Devlin et al., 2019), they do not easily slot into a broader

[3] Note that `fastText` provides a discontinued older version of their embeddings solely trained on Wikipedia corpora. While we have not formally determined how our pre-trained embeddings compare to those original `fastText` embeddings, we are confident that our versions are of comparable, if not higher quality. We decided to train our own Wikipedia-based `fastText` embeddings instead of relying on the original release for several reasons: (1) the `fastText` project indicated that their Wikipedia-only embeddings underperform compared to their Wikipedia+CommonCrawl embeddings; (2) it remained unclear what Wikipedia corpus `fastText` had used for their version; (3) we could not obtain the specifications and parameters used for preprocessing and training by `fastText`, which complicated a satisfactory performance of the corresponding ALC embeddings.

[4] See SI C and the SI K.

regression-style inference model with standard errors, $p$-values, etc.

### 1.3 Coverage and Intended Use

At the time of writing, we make all required products available for 40 of the most common languages (other than English).[5] This covers the majority of first and second-language speakers on Earth and the great majority of all languages on the web. Moreover, we have constructed pipeline production code for anyone who wishes to produce similar items for any of the 157 languages originally provided via `fastText`.

Our materials are aimed at two—often overlapping sets—of *low resource* users. First, analysts who work with languages that have relatively small corpora from which it is hard to learn high-quality embeddings. For example, scholars with a few political pamphlets or tweets from France may struggle to build embeddings for a relatively new term like "iel" (a gender-neutral pronoun) from such a small corpus. The alternative strategy—of translating the small corpus to a language for which embeddings do exist—may be unpalatable. Second, analysts who do not have local access to the computational resources required to train embedding models—we mean this both in terms of time/skill and power *per se*.

We now validate these approaches and discuss their relative performance. We first show that the ALC representations work well relative to the "full" embeddings that they approximate. We then focus high-cost efforts (i.e., crowdsourcing) on comparing (1) our version of `fastText` (fit to Wikipedia) against the original version of `fastText` and then (2) our version of `fastText` against an ALC version of our `fastText`. We do this because the `fastText` resources are the most innovative part of what we provide.

## 2 Performance And Validation

The resources we provide are useful to the extent that they provide reasonable representations of concepts, especially political ones. We now show that this is the case.

[5] We continue to expand the resources to additional languages using our training pipeline after publication.

## 2.1 Reconstruction: ALC Embeddings Provide Reasonable Approximations of the “Truth”

Recall that ALC embeddings are an *approximation* to (what we might describe as) true ones, where “true” means the embeddings estimated from a vast corpus. We have the latter insofar as we can learn `fastText` or `GloVe` embeddings from, say, Wikipedia. We can then compare that truth to our estimate (our ALC embedding). We would hope that our ALC embedding can reconstruct that truth and, on average, be “close” to it rather than “far” from it. These standards are vague in an absolute sense, but they do allow us some comparison across languages. The unit of comparison here is 100 random terms per language, constrained to have a higher frequency than the median token in the corpus.[6] For each term and each language, we estimate the cosine similarity between its pre-trained embedding and its corpus-wide ALC embedding. In SI E we describe exactly how this test proceeds.

The cosine similarities by construction range between $-1$ and 1. If this number is 1, then the ALC embeddings (of our random terms) perfectly approximate our “true” embeddings; if they are zero or even negative, they provide a very poor approximation. In Figure 1, we report the results for all the languages we have worked with so far, including the mean (diamond) and the cosine for each of the 100 random terms (circles).

[6] We make this restriction mainly to ensure that terms are actually in the relevant language. Especially for smaller languages, lower-frequency terms are often loan words in English/other languages. In Figure 14 of SI I, we illustrate that we receive similar results with terms at the 25th percentile of the type distribution in the vocabulary for larger languages.

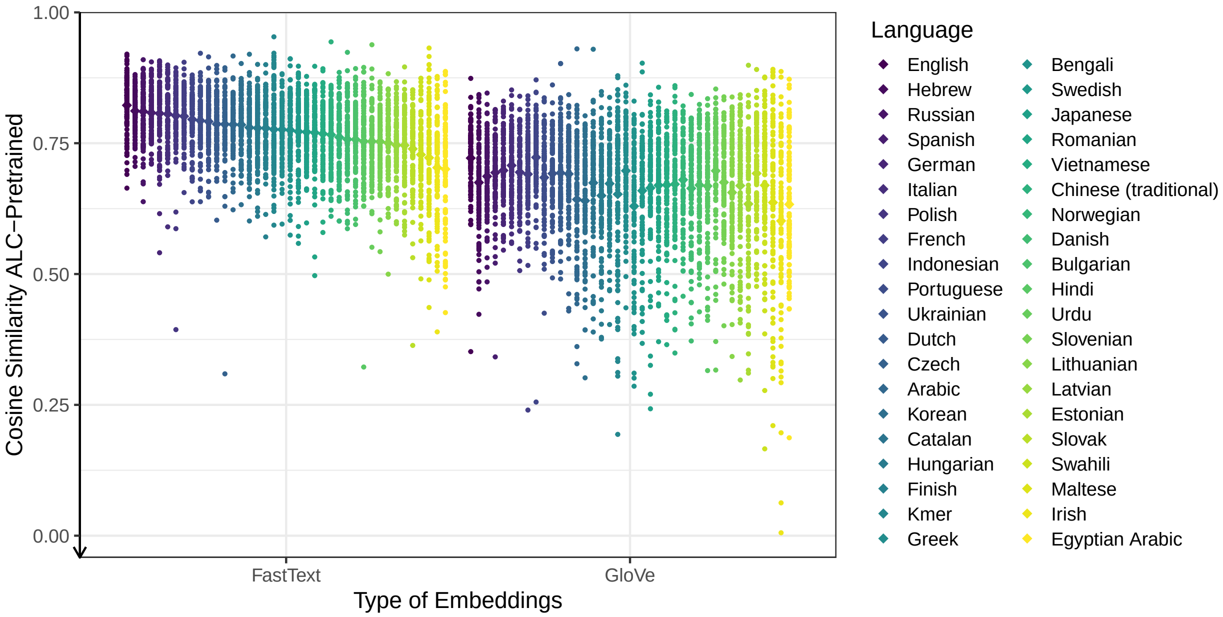


Figure 1: Reconstruction performance: cosine similarity between our ALC version of `fastText` and `GloVe` and those underlying architectures. Languages are ordered according to the mean accuracy for `fastText`. In theory, cosine similarities range between $-1$ and 1, but empirically all estimates are positive.

We have two immediate observations: First, ALC generally recovers both architectures' pretrained embeddings very well for any language. In general, means are around 0.77 for `fastText` and 0.67 for `GloVe`.[7] Second, there is non-trivial variation within and between languages. In particular, and as we show more explicitly in Figure 3 of SI D, ALC does best when there is more training data—for example, English has a higher mean than Irish. Moreover, within languages with lower means, we see longer left tails—that is, there are more terms further from the mean where ALC does a worse job of approximating the "truth". Again, this is primarily a consequence of training data availability.

A more qualitatively informative procedure is to check that words represented via our embeddings "mean" what we expect them to. We first verify this by studying a curated domain setting—specifically, translated English/Spanish speeches at the European Parliament (EP), 1999–

[7] It is very difficult to make firm comments comparing within language, across models (e.g. `GloVe` v `fastText` for German). This is because the accuracy is with respect to a within-architecture baseline (`GloVe`-ALC to `GloVe`; `fastText`-ALC to `fastText`), and assumes *a priori* that the analyst seeks to model the text specifically as that architecture does.

2001 (Høyland, Sircar and Hix, 2009). We proceed as described in SI F.

## 2.2 Crowdsourcing: Similar Aggregate Performance, ALC Delivers More Substantive Connotations

Another and somewhat easier way to assess the quality of our embedding resources in different languages is to look at the nearest neighbors of certain political terms. Consider Table 1. There, we provide nearest neighbors (by cosine similarity) for the terms `democracy` and `equality`. The nearest neighbors are drawn from two resources: our recompiled version of `fastText` and our ALC-based version of `fastText`.[8] Consistent with our notes above, the training corpus is (English) Wikipedia.

| **democracy** | | **equality** | |
|---|---|---|---|
| our fT | our fT-ALC | our fT | our fT-ALC |
| democracy | democracy | equality | equality |
| democracy's | democratising | equalities | non-discrimination |
| democracies | democracy's | non-discrimination | inclusiveness |
| democratization | internationalism | anti-discrimination | antidiscrimination |
| social-democracy | parliamentarism | anti-discriminatory | anti-discrimination |

Table 1: Nearest neighbors for English terms `democracy` and `equality`.

The good news is that these nearest neighbors make sense—that is, neither model produces "odd" results. Arguably, by moving beyond lexical similarities and similar word stems, ALC produces slightly more "useful" results than the pure `fastText` model. The same is true when we analyze the French terms `nationalisme` (nationalism) and `racisme` (racism), for which the training corpus is French Wikipedia, per Table 2.

[8] In Tables 5 and 6 in SI I, we repeat this exercise while further restricting nearest neighbors to terms that do not share the same word stem as the keyword. Evidently, both our `fastText` embeddings and our ALC-based version of `fastText` return meaningful nearest neighbors for political terms—beyond just lexical similarities.

| **nationalisme** | | **racisme** | |
|---|---|---|---|
| our fT | our fT-ALC | our fT | our fT-ALC |
| nationalisme | nationalisme | racisme | racisme |
| nationalismes | l'internationalisme | racismes | l'antiracisme |
| néonationalisme | internationalisme | antiracisme | communautarisme |
| régionalisme | radicalisme | l'antiracisme | antiracisme |
| internationalisme | néonationalisme | l'homophobie | l'islamophobie |

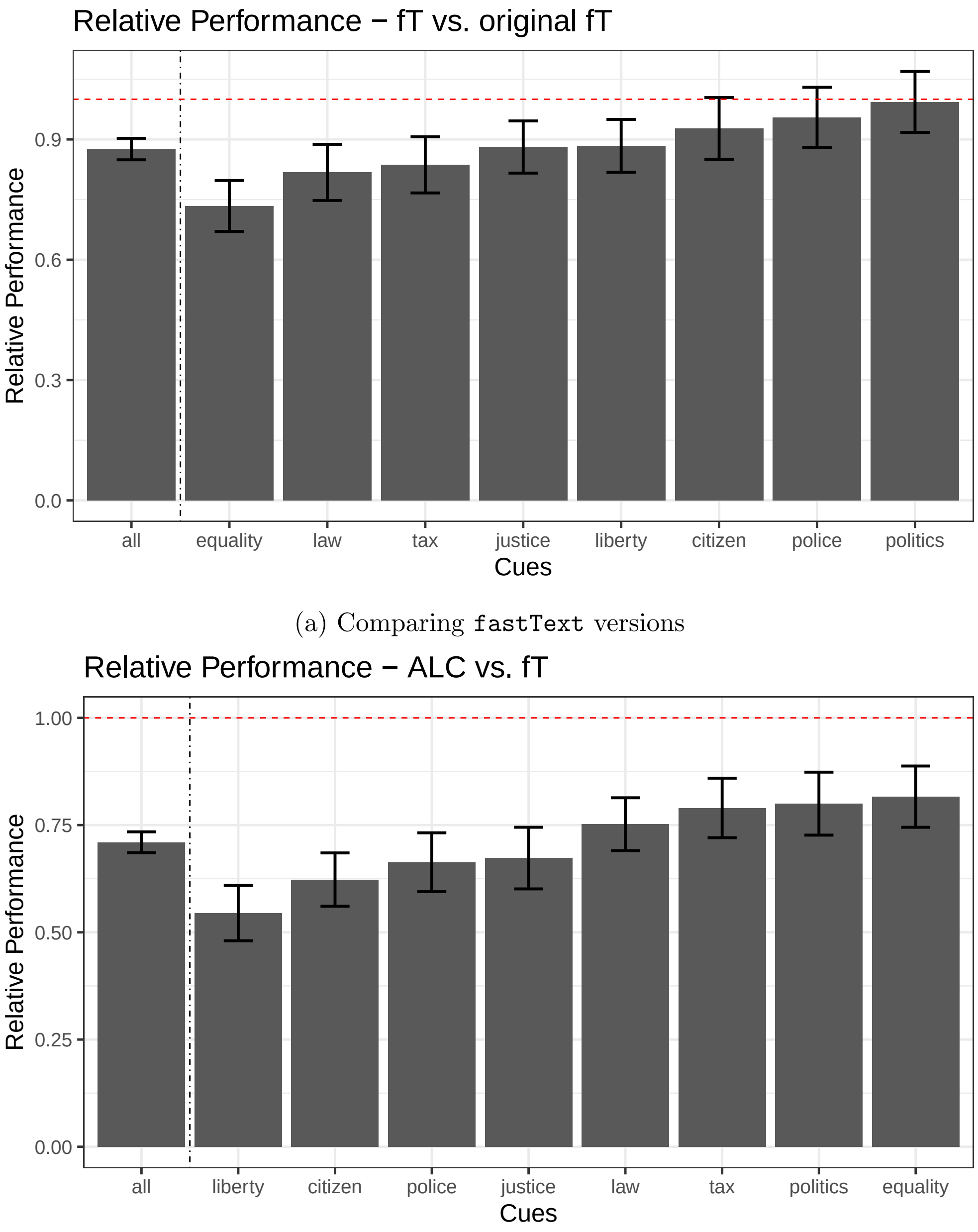
Table 2: Nearest neighbors for French terms `nation` and `racisme`.

To scale these comparisons between models, we turn to crowdsourcing (Benoit et al., 2016). Following Rodriguez and Spirling (2022), we designed a lightweight web application that shows crowdworkers a token with political connotations and then asks which of two words (drawn from two models) the worker thinks is a more plausible "context" term for that token. We translated the app into all of the (non-English) United Nations "Official Languages" and, in each language, we use eight 'political' terms (`law`, `liberty`, `equality`, `justice`, `politics`, `tax`, `citizen`, `police`). Hence, we evaluate Arabic, (traditional Mandarin) Chinese, French, Russian, and Spanish. In addition, we also created Japanese and Korean versions. If we take Rodriguez, Spirling and Stewart (2023) as sufficient evidence for the merits of ALC in English, then, combined with our exercise, we "cover" around 45% of the world's first and second languages and around 77% of the web's content languages.[9] Locating native speakers of these (non-English) languages was not trivial (and not cheap) in some cases. We worked with a specialist crowdsourcing firm, *CloudResearch*, for this purpose. In SI G we give more details on this process.

We ask crowdworkers to make two sets of comparisons: original `fastText` vs our version and then our version of `fastText` vs an ALC version of that resource. In Figure 2 we give an overview of the results. In the top subfigure, we report the comparison of our version of `fastText` to the original `fastText`. Each bar represents a term in the task (the far left bar is an overall result); we also include 95% confidence intervals. When that bar is higher than 1, respondents (on average) preferred our version; when below 1, they preferred the original. Ultimately, this comparison is

[9] See, e.g., https://w3techs.com/technologies/overview/content_language.

equivocal, with the original `fastText` being preferred in a couple of cases, but mostly, the difference is not statistically significant. The bottom subfigure compares our `fastText` to our ALC. Here, we see that, for the crowdworkers, ALC is generally not the preferred option, though again, this is equivocal in some cases.



(a) Comparing `fastText` versions

(b) Comparing `fastText` and ALC

Figure 2: Summary of crowdsourcing comparisons, all languages.
Baseline is original fT (in figure (a)) and fT (in figure (b)).

Across languages, crowdworkers mostly do not see huge differences in quality and have a mild preference for the (original) `fastText` resources (see SI H).[10] So does this mean an analyst should always prefer the original `fastText` over our version, including the one using ALC? The answer is 'no' for two reasons. First, the ALC embeddings give one access to the inferential machinery we discussed above. That is, the ALC embeddings are, by construction, an approximation, but they also allow one to conduct regressions, do statistical tests, etc. Second, and perhaps more fundamentally, these contest results disguise some important heterogeneity in use cases. Put simply, crowdworkers prefer more obvious "everyday" or "vanilla" nearest neighbors, whereas our new resources are likely helpful to analysts interested in technical terms. To see this concretely, consider Arabic—specifically, the Arabic word for `law`, قانون. The ALC nearest neighbor is المشرع (`legislator`), whereas the `fastText` nearest neighbor is قانونًيا (`legally`). Going down the list, `fastText` returns many lexical neighbors like قانونى (`legal`) and قانونه (a combination of a function word and the original keyword). Meanwhile, ALC returns more context-specific terms like الإلزام (`binding`) and التشريع (legislation).

A final note on our crowdsourcing data is that the comparisons were based on minimal preprocessing and post-processing of the embeddings. For example, we imposed only very small minimum counts for a given term to be included in their set of embeddings, specifically a minimum frequency of 10 occurrences in the language-specific Wikipedia corpus. We did this to make the comparison as 'raw' and clear as possible. However, following some internal experiments, we adjusted the various cut-offs upwards in our distributed resources. We did this especially for larger languages to ensure more robust and sensible embeddings. Put otherwise, the relative ALC vs. non-ALC crowd

[10] There is a subtlety to interpreting the results here: note that the ALC embeddings are simply averaged over the entire corpus (on which the `fastText` embeddings are themselves trained). That is, the 'context' of the ALC embeddings is the whole corpus, whereas they are actually designed, and should be optimal, for much more local use.

comparisons above are likely the worst-case scenario for ALC.[11]

## 3 Advice to Researchers using Our Resources

Our observations about ALC above are with reference to the relevant transformation matrix (**A**) having been estimated from the underlying corpus—specifically, Wikipedia. Unsurprisingly, whether this is appropriate for a given problem is a function of how 'close' the researcher's corpus is to Wikipedia. Here are three gradated scenarios to guide researchers in making such choices in practice:

1. Approximately in sample: if the researcher's local corpus is "close enough" to Wikipedia, then using our pre-fitted transformation matrix will work as well as anything else from the perspective of producing ALC embeddings. We demonstrate this with an example in SI J, where we use ALC embeddings for the German Wikipedia to identify homonyms.

2. Out of sample, small corpus. The researcher is out of sample if their corpus does not particularly resemble Wikipedia. If their corpus is too small to fit local models, we recommend using our estimated **A** matrix and carefully checking its validity. We give an example for this case using French and Italian parliamentary corpora in SI K.

3. Out of sample, large corpus. If their corpus is large, we advise researchers to simply fit a local transformation matrix using our pipeline code—and potentially fit their own embeddings. Of course, this involves a judgment call: the user must decide whether their inferences are better with our **A** for the language and corpus at stake or with their own (and/or with their own local embeddings). We did local fitting of **A** to our various parliamentary corpora to provide calibration. As illustrated in SI K, the results are satisfactory for the *Congressional Record* (median speech length 215 words) but unsatisfactory for the French and Italian corpora (median speech lengths 40 and 140 words, respectively).

[11] To reiterate, we provide full pipeline code such that users can recreate the resources under any pre or post-processing regime they wish.

To the extent researchers seek more concrete advice, our evidence suggests using our estimated quantities as a first cut on the problem. If they seem suitable and can be validated—for example, via substantive inspection of the nearest neighbors—then one can build out from there. If they do not seem suitable, consider estimating your own with our code. Subsumed in this recommendation is the idea that one might train with something other than Wikipedia on quality grounds. That is, we acknowledge that this resource has some plausible heterogeneity across languages, and analysts should use their expert judgment in deciding whether it is appropriate for their use case. In any case, our resources are a reasonable comparison point for any such work.

## Acknowledgements

First version: March 17, 2023. This version: July 17, 2024. A previous version of our work won the "Best Virtual Poster" award at the Summer Polmeth Meeting (2022). Christopher Lucas, Sebastian Popa, Clara Suong, and Luwei Ying provided very helpful comments on an earlier draft. We thank Mikhail Khodak for providing us with helpful feedback. We received excellent research assistance and advice from Dias Akhmetbekov, Alia ElKattan, Tatsuya Koyama, Cristina Mac Gregor Vanegas, Francis William Touola Meda, Yinxuan Wang, and Kyu Sik Yang.

## Data Availability

The resources discussed in this paper are here: http://alcembeddings.org/. This includes the training pipeline, the trained resources, and data. Replication code for this article has been published in Code Ocean, a computational reproducibility platform that enables users to run the code and can be viewed interactively at https://doi.org/10.24433/CO.1866319.v3 (Wirsching et al. (2024)).

## References

Benoit, Kenneth, Drew Conway, Benjamin E Lauderdale, Michael Laver and Slava Mikhaylov. 2016. "Crowd-sourced text analysis: Reproducible and agile production of political data." *American Political Science Review* 110(2):278–295.

Caliskan, Aylin, Joanna J. Bryson and Arvind Narayanan. 2017. "Semantics derived automatically from language corpora contain human-like biases." *Science* 356(6334):183–186.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*.

Gennaro, Gloria and Elliott Ash. 2022. "Emotion and reason in political language." *The Economic Journal* 132(643):1037–1059.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin and Tomás Mikolov. 2018. "Learning Word Vectors for 157 Languages." *CoRR* abs/1802.06893.

Høyland, Bjørn, Indraneel Sircar and Simon Hix. 2009. "Forum section: an automated database of the european parliament." *European Union Politics* 10(1):143–152.

Khodak, Mikhail, Nikunj Saunshi, Yingyu Liang, Tengyu Ma, Brandon Stewart and Sanjeev Arora. 2018. "A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors." *CoRR* abs/1805.05388.

Mikolov, Tomás, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and their Compositionality." *CoRR* abs/1310.4546.

Rheault, Ludovic and Christopher Cochrane. 2020. "Word Embeddings for the Analysis of Ideological Placement in Parliamentary Corpora." *Political Analysis* 28(1):112–133.

Rodman, Emma. 2020. "A Timely Intervention: Tracking the Changing Meanings of Political Concepts with Word Vectors." *Political Analysis* 28(1):87–111.

Rodriguez, Pedro L. and Arthur Spirling. 2022. "Word embeddings: What works, what doesn't, and how to tell the difference for applied research." *The Journal of Politics* 84(1):101–115.

Rodriguez, Pedro L., Arthur Spirling and Brandon M. Stewart. 2023. "Embedding Regression: Models for Context-Specific Description and Inference." *American Political Science Review* pp. 1–20.

Wirsching, Elisa M., Pedro L. Rodriguez, Arthur Spirling and Brandon M. Stewart. 2024. "Replication Data for: Multilanguage Word Embeddings for Social Scientists: Estimation, Inference, and Validation Resources for 157 Languages.". https://doi.org/10.24433/CO.1866319.v3.