Embedded Lexica:

Extracting Keywords from Unlabeled Corpora using Word Embeddings

Patrick J Chester*

University of California, San Diego

July 11, 2024

Abstract

Researchers frequently need to extract information, such as events or target topics, from large corpora. One common solution involves applying semantically-related keywords to identify tweets, news articles, or other documents of interest. However, it is rarely the case that dictionaries of relevance to the topic, event, or language both exist and are accessible. Existing algorithms for extracting dictionaries, require many user-provided seed words or hand-coded documents to generate useful results. Additionally, they do not incorporate contextual information from natural language. In this paper, I present a novel algorithm, `conclust`, that extracts keywords from unlabeled text using a small number of user-provided seed words and a fitted word embeddings model. Compared to existing methods of lexicon extraction, `conclust` requires few seed words, is computationally efficient, and takes word context into account. I describe this algorithm's properties and benchmark its performance with existing methods of lexical dictionary extraction, comparing differences in user labor, conceptual clarity, and the ability to replicate existing keyword dictionaries.[1]

*Patrick Chester is a postdoctoral researcher at the UC, San Diego China Data Lab. Please relay questions and comments to his email address: patrickjchester@gmail.com. Links to his research and software can be found at his website: www.patrickjchester.com.

[1]I would like to acknowledge and thank UCSD Professor Margaret Roberts for providing the labor and financial resources needed to produce this paper. I would also like to thank Professor Arthur Spirling for comments and suggestions that helped improve this paper.

Introduction

In the social sciences, there is a significant mismatch between the supply and the demand for keywords. Researchers who perform sentiment analysis, machine learning, and data exploration with text data frequently find themselves in need of high-quality dictionaries that are relevant to their research questions and are appropriate fits for the text in their possession.

Many researchers have responded to this problem by using pre-defined keywords sets. However, this approach is ill-advised as the semantics of words can differ dramatically across different contexts and corpora (Quinn et al. 2010). Alternatively, many researchers have compiled targeted keyword sets using a combination of intuition and subject knowledge. While this method is capable of producing valauble and relevant outputs, it is too costly and time-consuming to be practical in most use-cases. Existing semi-supervised methods for producing dictionaries can involve a significant amount of human labor either to identify conceptually related terms or to hand code documents for the consumption of a machine learning algorithm, such as the model described by King, Lam, and Roberts (2017).

Unsupervised machine learning methods offer some promising options to address this need. In particular, word embeddings, numeric representations of the semantic meaning of words, have been used to produce sentiment dictionaries and compare how concepts are associated in text (Rice and Zorn 2021). One advantage of the embeddings-based approach is that it is possible for models to incorporate semantic information from domain-specific corpora. Additionally, they require minimal input from researchers to generate high quality results. In sum, they represent a potential improvement in both the efficiency and quality of keyword production over alternative methods.

In this paper, I extend existing work that applies word embedding methods to generate sentiment dictionaries to the more general objective of producing conceptual dictionaries, also known as keywords (King, Lam, and Roberts 2017). I do so using a novel algorithm called `conclust`, which takes seed words as an input and produces a set of keywords that

are semantically similar to one another and the seeds.[2] Using the Turing test approach, I use a set of human-coded terms to examine how `conclust` performs over several parameter specifications (Turing 2012; Spirling and Rodriguez 2021). Additionally, I benchmark its performance against a well-documented set of conceptual dictionaries produced by the WordNet project (Miller 1995).

This paper is structured as follows, first I review the methods that have been used to generate keywords in the past, examining their advantages and limitations. Second, I describe the `conclust` algorithm, the inputs it requires and its properties. Third, I compare dictionaries produced by `conclust` with labeled terms to identify how to best apply it. Finally, I validate this approach by comparing `conclust` dictionaries with those produced by more traditional methods.

Literature

Keyword and conceptual dictionaries have been a part of social scientific research for decades. They are used in a variety of ways: identifying documents relevant to event extraction (Goldstein and Pevehouse 1997); topic labeling, extraction, and analysis (Laver and Garry 2000); and as target and attribute words for embedding analysis (Yang and Roberts 2021; Chester 2023).

Currently, keywords tend to be generated using three distinct approaches. First, human coders are frequently used to compile and validate keywords. This approach has some advantages, including its incorporation of human judgement into the dictionary generation process. This is the procedure used to generate many well documented and validated semantic dictionaries, such as WordNet (Miller 1995; Fellbaum 2010). On the other hand, it is quite costly to implement, which is a potentially significant barrier to entry for smaller research operations.

The second common method for generating keywords is to use supervised machine learn-

[2]This algorithm has been implemented in conclust an R package that is available on github.

ing methods (King, Lam, and Roberts 2017). The benefit of this approach is that it can be customized to specific subcorpora and is significantly less costly to implement compared to using human coders. That said, this process still requires both the provision of human-provided seed words and documents that are hand-coded by researchers.

Third, scholars, such as Häffner et al. (2023), have explored using deep learning as a tool for generating dictionaries. Their approach leverages the weights of a fitted deep learning model and their association with a continuous outcome variable to identify words whose weights are predictive of the target topic. It bears some similarities to that of King, Lam, and Roberts (2017), though the weights of a neural network model have the advantage of representing non-linear relationships between text and an outcome of interest. However, the utility of this methodology is largely limited to cases where a large corpus is paired with a variable of interest to the researcher.

Finally, in recent years, researchers have increasingly turned to word embeddings as a tool for accomplishing a similar task: the creation of sentiment dictionaries. To do so, they leverage a core feature of word embedding methods, that they generate word vectors that represent the semantic meaning of words as they appear in a given text corpus. They take seed words that represent opposite poles of a sentiment spectrum, a fitted word embedding model, and identify words that lie on a continuum between the chosen seed words (Rice and Zorn 2021). It's worth noting that thus far embedding-based methods have been limited to the specialized task of creating polarized sentiment dictionaries. In contrast, keywords do not have any polarity and can represent nominal concepts, such as "politics," "science," or "ethnicity."

conclust

The `conclust` algorithm is designed with a core question in mind: how can we produce semantically-related keywords in a way that is labor and cost-efficient, is customizable to

specific languages or text corpora, and is reproducible by other researchers? In this section I describe the design of the algorithm, its features, and the ways in which it can be applied to add.

This algorithm is designed to replicate the advantages of the embedding-based model developed by Rice and Zorn (2021) to generate semantically-related sets of keywords. In their words, embeddings have the potential to make the process of generating keywords more *generalizable* and *efficient*. This is because keywords produced by embeddings are generalizable, as they can be applied to any corpus of interest to the researcher. The process is also highly efficient compared to using supervised methods or human production, as embeddings do not require human labor to produce and they can produce results with minimal inputs. Whereas Rice and Zorn (2021) applied embeddings to produce sentiment dictionaries, I hope to apply this approach towards the creation of more general keyword sets that represent concepts of value to researchers.

What value do embeddings add to the production of sets of keywords? The main benefit is that high-quality fitted word embedddings model contains word vectors that can be used to represent the semantic meaning of words given the corpus that model was fitted upon. These semantic meanings can be used to identify similarities between words given the contexts in which they appear (Mikolov et al. 2013). The `conclust` algorithm (see Algorithm 1) leverages these semantic word vectors to obtain the set of keywords that are iteratively most similar to a user-provided set of seed words.

`Conclust` requires several inputs: seed words, a fitted embedding model, and user provided size and similarity thresholds. In this context, *seed words* refer to a set of user-provided words that represent the target concept. Typically, they vary in size from two to eight words, with larger sets of seed words generally increasing the likelihood that the target concept will be represented in conclust output. The fitted embedding model can be represented as a $n \times m$ matrix where n is the number of tokens in the fitted model and m is the number of

embedding dimensions.[3] Finally, the model takes two user inputs that shape the model's stopping point. The size threshold is the maximum number of tokens that can be output from the conclust model. The similarity threshold, t, indicates the minimum average cosine similarity to the current set of dictionary words that a new word must have to be added to it. Higher thresholds ensure that the resulting dictionary will be smaller but more co-similar; lower thresholds will do the opposite.

Algorithm 1: `conclust`

Input: Seed words: S; Embedding model M; Size threshold: n;
Similarity threshold: t
Result: Keyword set: K
$K = S$;
while $|K| \geq n$ **do**
  $\bar{m} = \forall m \in M$ `max(sim(`K, m`))`;
  if *mean(sim(K,* $\bar{m}$*))* $\geq t$ **then**
    $K = K \cup \bar{m}$;
  else
    break;
  end
end

When `conclust` is provided these inputs, it computes the cosine similarity between the seed word set and the remaining words in the fitted word embeddings model. The word that has the highest average similarity to the seed set, $\bar{m}$, is identified and added to that set. This process continues until either the size (n) or similarity thresholds (t) are met. The end results of this process is a set of words that are iteratively co-similar in semantic meaning. This process is deterministic, so that given the same inputs the model will generate identical results.

[3] `Conclust` does not require that the embedding model be produced by any specific model. Thus far, I have experimented with *word2vec*, *GloVe*, and *FastText* models and they have all performed comparably well. More important than the model type is the size and quality of the data upon which they were fitted (Mikolov et al. 2013; Pennington, Socher, and Manning 2014; Bojanowski et al. 2017).

Evaluation

When discussing any new text as data method, our focus should be first validating that it works as intended and is an improvement on existing and commonly used alternatives (Quinn et al. 2010; Grimmer and Stewart 2013). In this section I validate `conclust` in several ways. First, I present sets of keywords produced by `conclust` that are designed to represent concepts that could plausibly be of value to users. Second, I describe how I use human subjects to generate comparison keyword sets with which `conclust's` keywords can be compared. Third, I examine how the quality of `conclust's` keywords varies over various relevant parameters: the number of seed words provided and the keyword size threshold.[4] Finally, I examine how `conclust's` keyword sets compare to those produced by human coders at the WordNet project (Miller 1995).

To evaluate the quality of keywords produced by `conclust`, it is necessary to have a reference group of terms that both are and are not relevant to target concepts. To create these comparison words, I utilize human coders.[5] Why human coders? Despite the progress that has been made in the application of machine learning to the generation of sentiment and conceptual dictionaries, human coders remain a popular tool in the generation of high-quality targeted dictionaries. Accordingly, human-coded dictionaries represent an effective benchmark against which dictionaries can and should be compared. However, there is no question that there is a degree of subjectivity in the decision to assign a word to a concept, even among the best trained research assistants. Concurrently, this is the rationale for including WordNet dictionaries (see Table 1) as a comparison group. Ideally, `conclust` will produce dictionaries that are of are rated to be of comparable quality to WordNet by the human evaluators. Should human evaluators be indifferent or even prefer the `conclust` dictionaries to those of WordNet, then one could argue that they pass the Turing test as it

[4] I will analyze the impact of changes to the similarity threshold in a future draft of this paper.

[5] Two undergraduate students at University of California, San Diego were responsible for producing the human-coded terms used to evaluate the `conclust` and WordNet keywords.

applies to text as data analysis (Turing 2012; Spirling and Rodriguez 2021).[6]

Table 1: Full Set of WordNet Keywords

ID	biology	economy	executive	government	sport
1	biology	economy	director	government	sport
2	science	market	business	regime	rock
3	botany	enterprise	chairman	state	contact
4	ecology	capitalism	board	bureaucracy	field
5	space	capitalist	chief	court	exercise
6	forestry	venture	officer	empire	track
7	microbiology	socialism	ceo	commission	water
8	biotechnology	socialist	operating	plan	row
9	biotech	communism	cfo	town	archery
10	engineering	nazism	insider	meeting	horseback
11	recombinant	mercantile	president	palace	cycling
12	dna		minister	puppet	blood
13	technology		government	welfare	game
14	morphology		cabinet		judo
15	anatomy		chancellor		spectator
16	topology		secretary		team
17	neuroscience		home		boxing
18	brain		state		wrestling
19	physiology		lord		golf
20	zoology		treasury		football
21	shell		finance		baseball
22			surgeon		basketball
23			vice-president		tennis

Note:
All available WordNet tokens for each concept were included with the exception of entries that included more than one token.

Each human coder was provided with a set of five concepts and a list of terms 2800 tokens. The tokens included a mixture of terms that had been identified by `conclust`, WordNet, and some that were randomly selected from the same word embedding model used to by conclust.[7] The human coders were instructed to assign these tokens into one or more of

[6]I also evaluate the performance of human coders against a random subset of 100 words that were labeled by the author, i.e., a gold standard set of words. The human coders have performed reasonably well against this benchmark, achieving an F1-score of 0.75 relative to the gold standard.

[7]The inclusion of randomly selected terms was done to determine what types of conceptually-relevant

these five categories, or a neutral category if none of the concepts provided were a close match. Additionally, the human coders were provided definitions of the respective concepts and instructed to be conservative in their allocation of words to concepts; i.e. they were told that type 2 classification errors were preferable to type 1. The rationale for this instruction is to minimize noise caused by the misclassification of irrelevant terms. Additional details about the instructions provided to human coders are provided in the Appendix.

To examine how the quality of keywords sets generated by `conclust` varies over various model inputs, I created a set of five seed words (see Table 2) for each of the five target concepts: *biology*, *economy*, *executive*, *government*, and *sport*.[8] The `conclust` algorithm generated a separate keyword set for each combination of seed words for each concept for a total of 155 separate sets of 50 keywords.

Table 2: Seed Words Used to Generate Conceptual Dictionaries

biology	**economy**	**executive**	**government**	**sport**
biology	economy	president	government	sport
dna	gdp	ceo	policy	baseball
organism	capital	manager	law	football
evolution	job	chairman	legislator	ball
phenotype	investment	minister	president	tennis

The word embedding model used by `conclust` to generate dictionaries was the pre-trained *FastText* model that was fitted on the English version of Wikipedia and data from the Common Crawl Project (Grave et al. 2018). This model was selected largely because it is algorithm is efficient enough to run on most laptops, it is designed to generate high quality embeddings even for rarely occurring words, and Meta has produced pre-fitted FastText models for 157 languages (Bojanowski et al. 2017).

Pre-trained models are typically fit with minimal to no pre-processing and this was also

tokens were missed by both WordNet and `conclust`. It also provides a means by which recall may be computed for cross-keyword set comparisons.

[8]These concepts were selected to represent several distinct domains of knowledge and they each have corresponding dictionaries of at least 10 words from WordNet (see Table 1). The seed words were selected to represent the author's understanding of these concepts.

the case for the FastText models. As a consequence, the model included vectors for multiple tenses and forms of root words, as well as punctuation, and stop words. As the goal of any keyword generator should be to generate keyword sets with minimal redundancy and waste, I droped word vectors from the model that were associated with punctuation and stop words, and averaged vectors for tokens that included different types of capitalization or shared a root lemma. Additionally, as WordNet uses nouns exclusively in each of its keyword sets, I also remove all non-noun words from the FastText embedding model.

To compare keyword quality across various model parameterizations, I evaluate the degree to which users have correctly identified terms relevant to the target concepts using the precision, recall, and F1-score performance metrics. In this context, they each have interesting interpretations worth discussing. Precision is the ratio of true positives to the sum of true positives and false positives ($\texttt{TP}/(\texttt{TP}+\texttt{FP})$). In the context of evaluating keywords, it tells us the proportion of words that were identified by a given methodology that were relevant to a given target concept. Conversely, recall is the proportion of true positives to true positives and false negatives ($\texttt{TP}/(\texttt{TP}+\texttt{FN})$); here, this indicates what proportion of the broader set of relevant words were missed using a given keyword-generation method. Finally, the F1-score gives us the harmonic mean of precision and recall. High F1-Scores for a keyword set would indicate that they contain mostly relevant words and a high proportion of the total relevant words in the provided embedding model (see Equation 1).[9]

$$\frac{2 \times \texttt{Precision} \times \texttt{Recall}}{\texttt{Precision} + \texttt{Recall}} \tag{1}$$

How will the quality of keywords produced by `conclust` perform given varying number of seeds and over varying lengths of keyword sets? My expectation is that as the number of seeds the algorithm will have more information about the semantic cluster of interest to

[9]Of these performance metrics, I expect the one that will be most relevant to users is precision, as most users will likely prefer to use dictionaries that contain a minimal number of irrelevant words. Recall will be relevant for users who intend to identify as many conceptually relevant terms in a given corpus as possible, even if some of those terms are not relevant to the concept of interest.

the researcher; accordingly, the quality – both precision and recall – of keywords produced by `conclust` should monotonically increase with the number of seeds provided. That said, each keyword provided to `conclust` represents a unit of labor from users of the algorithm; therefore, we can say the algorithm is efficient if it we observe it achieve peak performance with a minimal number of seeds.

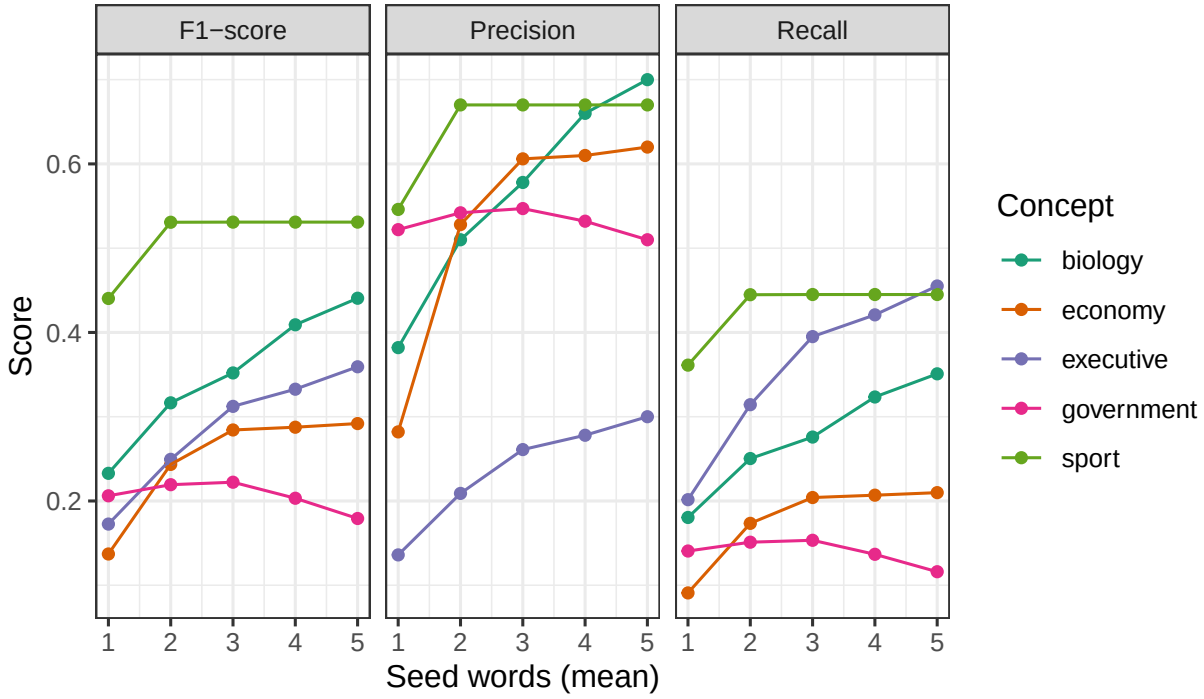
For the keyword length parameter, we would expect that recall will increase monotonically with longer keyword outputs from `conclust`. This follows the simple intuition that for every word included in a keyword set, there exists some non-zero probability that it is relevant to the user's target concept. However, if the algorithm works as intended, the first words added to the keyword set will be more relevant than words added later; therefore, as keyword set length increases, precision should decrease.

Results

In this section, I evaluate the quality of keyword sets generated by `conclust` relative to the evaluations made by human coders over the variation of several parameters: seed count and dictionary length. I then benchmark these dictionaries against the performance of relevant dictionaries from WordNet.

Below is Figure 1, which shows how the average F1-score, precision, and recall of keywords varies over the number of seed words used to produce them. Each point represents the average performance of keyword sets across all combinations relative to human keyword evaluations. For instance, the F1-score for the concept *biology* with two seed words represents the average F-1 score across all keyword sets generated by each possible pairing of the seeds presented in Table 2 column 1: [biology, dna], [biology, organism], [dna, organism], and so on. For reference, I also include the top 30 terms for each concept in Table 3 (the full set of 50 terms are included in the Appendix); the keyword sets in this table were produced by using all five seed words for each respective concept.

Figure 1: Conclust Performance over Seed Word Count



Note: This figure was produced by varying the number of seeds used by `conclust` to produce keyword sets, drawn from the population of seeds shown in Table 2. In each case where there was more than a single combination of seeds, the average across each combination is shown. They y-axis represents performance metrics of these keyword sets relative to evaluations made by human coders.

Across the three of the five concepts evaluated, there appear to be three general patterns. First, it is clear that for *sport* keyword sets, there is negligible improvement in the quality of keywords produced by two or more seeds. Coincidentally, keyword sets associated with this concept also have the highest overall F1-score compared to the four other concepts. Second, we see that the *biology*, *executive*, and *economy* show a gradual increase across all three measures of performance as the number of seeds increase. Finally, the *government* concept appears to increase in performance for the first two seeds, followed by a decline when four or five seed words were used. Additionally, the results shown in Table 3 indicate that my performance metrics are understating the quality of keyword sets produced by `conclust`; of the top 30 terms of each concept, few appear to be irrelevant to the target concepts.

Table 3: Top 30 Conclust Keywords Generated using All Five Seed Words

ID	biology	economy	executive	government	sport
1	recombinant	equity	commissioner	statute	soccer
2	mrna	mortgage	legislator	legislation	goalkeeper
3	protein	lender	elect	amendment	goalie
4	gene	finance	chairperson	mandate	championship
5	rna	security	governor	regulation	basketball
6	cdna	investor	committee	enact	player
7	peptide	liquidity	treasurer	constitution	coach
8	enzyme	creditor	mayor	enactment	preseason
9	mutation	debt	secretary	prohibition	playoff
10	kinase	loan	council	decree	volleyball
11	allele	financing	deputy	statutory	tournament
12	synthase	banking	vice-president	ordinance	scorer
13	molecule	asset	comptroller	stipulate	hockey
14	ligand	insolvency	delegate	provision	scrimmage
15	methylation	bank	appoint	state	teammate
16	chromosome	borrower	senator	enforce	postseason
17	biosynthesis	banker	politician	prohibit	midfielder
18	genome	holding	congressman	authority	softball
19	biochemistry	company	incumbent	ratification	team
20	polymorphism	debtor	officer	jurisdiction	handball
21	dehydrogenase	refinance	government	govern	tourney
22	receptor	issuer	councillor	declaration	goaltender
23	polymerase	repayment	director	authorize	lacrosse
24	tyrosine	shareholder	legislature	ratify	rookie
25	metabolite	valuation	supervisor	agreement	quarterback
26	pcr	income	adviser	clause	squad
27	plasmid	corporation	leader	amend	roster
28	histone	insurer	councilman	legislature	fullback
29	sequence	fund	lawmaker	treaty	matchup
30	actin	citigroup	re-election	obligation	midfield

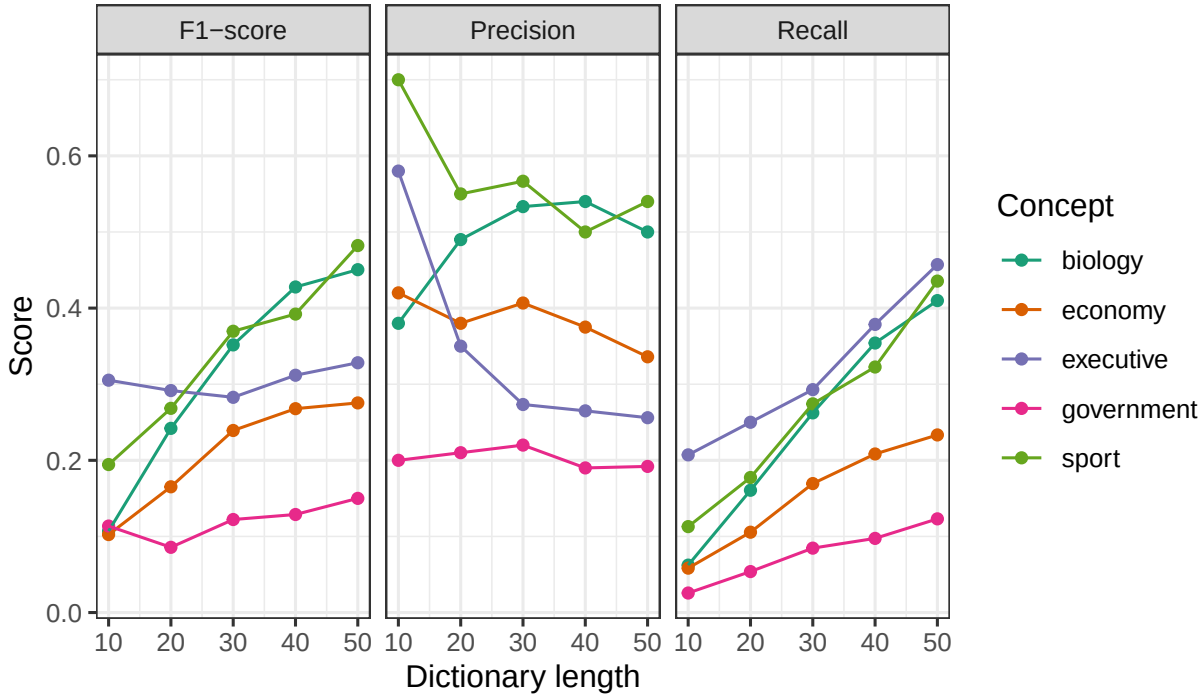
Note:
Each dictionary was generated by `conclust` using the five seed words shown in Table 2 for each respective target concept.

Overall, these results are generally consistent with expectations: as the number of seed words increases, the overall quality of the keywords produced by `conclust` increases. However, the precision of keyword sets appears to plateau in three out of five cases with three

seed words. This indicates that users of `conclust` may find little improvement in the overall quality of their dictionaries if they use more than three or four seed words. This suggests that `conclust` requires minimal information from users to produce quality results; i.e. it is efficient from a user's perspective.

The second parameter that we use to evaluate `conclust` is dictionary length. Generally, I expect that the longer a keyword set is, the smaller percentage of its words will be relevant (lower precision), yet the higher percentage of the total relevant words in the corpus will be included (higher recall). To test whether this is the case, I assess the performance of `conclust` over variable lengths of keyword sets (see Figure 2 below).[10]

Figure 2: Conclust Performance over Dictionary Length



Consistent with expectations, we see increases in recall and decreases in precision as dictionary length increases. However, the trends are not symmetric: while recall consistently across all five concepts, precision is relatively constant for *government* and it follows a concave

[10]To simplify the analysis of dictionary length, it was limited to the average performance scores of dictionaries produced using three seed words. The results for alternative seed words were largely consistent with this analysis.

pattern for *biology*. When examining the F1-score, for three out of five concepts' gains in recall are roughly counterbalanced by losses in precision when dictionaries are 30 to 40 elements in length. However, for *biology* and *sport*, F1-scores continue to increase even for keyword sets of up to 50 elements. This suggests that the optimal dictionary length is highly dependent on the target concept in question. For some highly complex concepts that include large numbers of relevant words, a researcher would benefit from setting very large keyword length thresholds. On the other hand, some concepts – such as *executive* and *government* – appear to be sparser, i.e. they have fewer relevant words, and thus they see have the highest F1-score at shorter dictionary lengths (10 or so elements).

In sum, when determining dictionary length, researchers should be mindful of the scope of the concept that they are targeting: is it a narrow topic or one that is multi-faceted? For narrow topics, researchers are likely better off setting lower thresholds of thirty or fewer words. On the other hand, when creating keywords for topics that are quite large in scope, researchers should feel comfortable setting the threshold considerably higher: at 50 words or more.

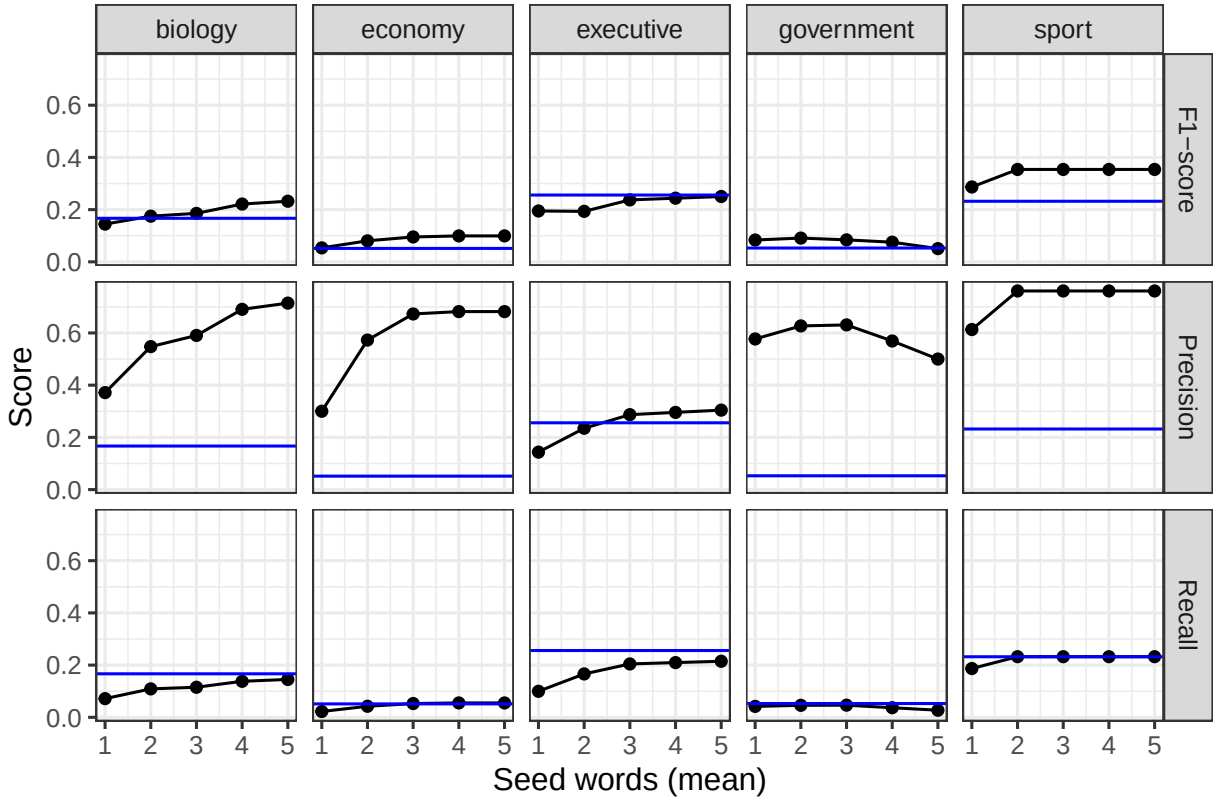
Validation with WordNet

Given our hand-coded data, how do the dictionaries produced by `conclust` perform relative to human-produced WordNet dictionaries? In Figure 3 each cell represents the precision, recall, or F1-score each of the five concepts; the horizontal blue line represents how the human coders evaluated the WordNet dictionaries, while the black line represents their evaluations of `conclust` at various seed levels.[11]

The results presented in Figure 3 suggest that in most cases `conclust` performs at least as well as WordNet according to human coders. As we saw before, we generally see performance of `conclust` increase with the number of seed words. In particular, when using three seed words, each concept has as least as high an F1-score as WordNet. Moreover, `conclust's`

[11]For each concept, the `conclust` dictionary was limited to size of each respective WordNet dictionary to ensure comparability.

Figure 3: Conclust Performance over Seed Length Compared to WordNet



economy and *sport* keyword sets appear to perform strictly better than WordNet's all seed counts.

When we decompose the F1-score, `conclust` dictionaries appear to perform particularly well according to the precision metric. Across all seed counts greater than three, `conclust` produces keyword sets that are in greater agreement with the human evaluations than those produced by WordNet. However, according to the recall metric, WordNet performs on par with `conclust` for *biology* and *government* concepts when five and three seeds are used, respectively. This suggests that while `conclust` is generally more likely to produce dictionaries that are consistent with the priors of independent human observers, researchers should experiment with their seed words to obtain dictionaries that capture the maximally relevant words from the corpus. Overall, it appears as though `conclust` produces dictionaries that perform at least as well as WordNet across most concepts, model configurations, and metrics, which suggests that it passes the Turing test.

Conclusion

In this paper, I have presented a novel algorithm based on word embeddings that can quickly and efficiently generate custom keywords for researchers. Given that it has a foundation in word embedding models, it has the advantage of generalizability, as it can be used to produce keywords specific to any corpus of sufficient size to fit a word embedding model. Should the researcher be uninterested in fitting their own embedding model, they could also use pre-trained embeddings models, as was done in this paper. The `conclust` algorithm is also highly labor efficient, as there is no requirement for researchers to label documents or to draw upon additional sources of data; instead, it can create high-quality keywords with 2-4 seeds provided by the user. I also evaluated my models across multiple model configurations, finding that while more seed words generally improves model performance, the optimal size of a keyword set is highly dependent on the target concept and how well represented it is in

the data. Finally, when compared to dictionaries generated by human coders, it performs at least as well, if not better according to blind human evaluations.

References

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. *Enriching Word Vectors with Subword Information* [in en]. ArXiv:1607.04606 [cs], June. Accessed April 5, 2023. http://arxiv.org/abs/1607.04606.

Chester, Patrick J. 2023. *Framing Democracy: Characterizing China's Negative Legitimation Propaganda using Word Embeddings.* Technical report. https://doi.org/10.7765/9781847794550.00007.

Fellbaum, Christiane. 2010. "WordNet" [in en]. In *Theory and Applications of Ontology: Computer Applications,* edited by Roberto Poli, Michael Healy, and Achilles Kameas, 231–243. Dordrecht: Springer Netherlands. ISBN: 978-90-481-8847-5, accessed April 10, 2023. https://doi.org/10.1007/978-90-481-8847-5_10. https://doi.org/10.1007/978-90-481-8847-5_10.

Goldstein, Joshua S., and Jon C. Pevehouse. 1997. "Reciprocity, Bullying, and International Cooperation: Time-series Analysis of the Bosnia Conflict" [in en]. *American Political Science Review* 91, no. 3 (September): 515–529. ISSN: 0003-0554, 1537-5943, accessed April 10, 2023. https://doi.org/10.2307/2952072. https://www.cambridge.org/core/product/identifier/S0003055400210940/type/journal_article.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. "Learning word vectors for 157 languages." *arXiv preprint arXiv:1802.06893.*

Grimmer, Justin, and Brandon M. Stewart. 2013. "Text as data: The promise and pitfalls of automatic content analysis methods for political texts." *Political Analysis* 21 (3): 267–297. ISSN: 14764989. https://doi.org/10.1093/pan/mps028.

Häffner, Sonja, Martin Hofer, Maximilian Nagl, and Julian Walterskirchen. 2023. "Introducing an Interpretable Deep Learning Approach to Domain-Specific Dictionary Creation: A Use Case for Conflict Prediction." Publisher: Cambridge University Press, *Political Analysis,* 1–19.

King, Gary, Patrick Lam, and Margaret E Roberts. 2017. "Computer-Assisted Keyword and Document Set Discovery from Unstructured Text." *American Journal of Political Science,* 1–49. ISSN: 15405907. https://doi.org/10.1111/ajps.12291.

Laver, Michael, and John Garry. 2000. "Estimating policy positions from political texts." Publisher: JSTOR, *American Journal of Political Science,* 619–634.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space.* Technical report. ArXiv: 1301.3781v3 Publication Title: arxiv.org. Accessed July 20, 2020. http://ronan.collobert.com/senna/.

Miller, George A. 1995. "WordNet: a lexical database for English." MAG ID: 2081580037, *Communications of The ACM* 38, no. 11 (November): 39–41. https://doi.org/10.1145/219717.219748.

Pennington, Jeffrey, Richard Socher, and Christopher Manning. 2014. "Glove: Global Vectors for Word Representation." *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP),* 1532–1543. https://doi.org/10.3115/v1/D14-1162. http://aclweb.org/anthology/D14-1162.

Quinn, Kevin M., Burt L. Monroe, Michael Colaresi, Michael H. Crespin, and Dragomir R. Radev. 2010. "How to Analyze Political Attention with Minimal Assumptions and Costs" [in en]. *American Journal of Political Science* 54, no. 1 (January): 209–228. ISSN: 00925853, 15405907, accessed March 10, 2023. https://doi.org/10.1111/j.1540-5907.2009.00427.x. https://onlinelibrary.wiley.com/doi/10.1111/j.1540-5907.2009.00427.x.

Rice, Douglas R., and Christopher Zorn. 2021. "Corpus-based dictionaries for sentiment analysis of specialized vocabularies." *Political Science Research and Methods* 9 (1): 20–35. ISSN: 20498489. https://doi.org/10.1017/psrm.2019.10.

Spirling, Arthur, and Pedro L Rodriguez. 2021. "Word Embeddings What works, what doesn't, and how to tell the difference for applied research." *Journal of Politics,* 1–56. https://www.nyu.edu/projects/spirling/documents/embed.pdf.

Turing, Alan M. 2012. "Computing machinery and intelligence (1950)." *The Essential Turing: the Ideas That Gave Birth to the Computer Age,* 433–464.

Yang, Eddie, and Margaret E Roberts. 2021. "Censorship of Online Encyclopedias : Implications for NLP Models." ArXiv: 2101.09294v1 ISBN: 9781450383097.

Appendix

Supplemental Tables

Table 4: Full Set of Conclust Keywords Generated using All Five Seed Words

ID	biology	economy	executive	government	sport
1	recombinant	equity	commissioner	statute	soccer
2	mrna	mortgage	legislator	legislation	goalkeeper
3	protein	lender	elect	amendment	goalie
4	gene	finance	chairperson	mandate	championship
5	rna	security	governor	regulation	basketball
6	cdna	investor	committee	enact	player
7	peptide	liquidity	treasurer	constitution	coach
8	enzyme	creditor	mayor	enactment	preseason
9	mutation	debt	secretary	prohibition	playoff
10	kinase	loan	council	decree	volleyball
11	allele	financing	deputy	statutory	tournament
12	synthase	banking	vice-president	ordinance	scorer
13	molecule	asset	comptroller	stipulate	hockey
14	ligand	insolvency	delegate	provision	scrimmage
15	methylation	bank	appoint	state	teammate
16	chromosome	borrower	senator	enforce	postseason
17	biosynthesis	banker	politician	prohibit	midfielder
18	genome	holding	congressman	authority	softball
19	biochemistry	company	incumbent	ratification	team
20	polymorphism	debtor	officer	jurisdiction	handball
21	dehydrogenase	refinance	government	govern	tourney
22	receptor	issuer	councillor	declaration	goaltender
23	polymerase	repayment	director	authorize	lacrosse
24	tyrosine	shareholder	legislature	ratify	rookie
25	metabolite	valuation	supervisor	agreement	quarterback
26	pcr	income	adviser	clause	squad
27	plasmid	corporation	leader	amend	roster
28	histone	insurer	councilman	legislature	fullback
29	sequence	fund	lawmaker	treaty	matchup
30	actin	citigroup	re-election	obligation	midfield
31	genotype	bankruptcy	trustee	sanction	striker
32	hemoglobin	brokerage	election	stipulation	champ
33	phosphorylation	broker	delegation	imposition	offseason

34	lymphocyte	treasury	candidate	issuance	footballer
35	mitochondrion	insurance	chief	issue	varsity
36	inhibitor	liquidation	leadership	exemption	premiership
37	subunit	market	auditor	permit	linebacker
38	apoptosis	business	congress	act	hitter
39	overexpression	firm	lobbyist	authorization	kickoff
40	transduction	liability	authority	constitute	shutout
41	chromosomal	subprime	administrator	revocation	scoreboard
42	recombination	annuity	administration	establishment	rugby
43	cytokine	dividend	presidency	enforcement	umpire
44	monomer	issuance	principal	assent	lineup
45	hybridization	bailout	spokesperson	congress	badminton

Note:
Each dictionary was generated using all five seed words for each respective concept as represented in Table 2

Research Assistant Training Materials

Research assistants were provided the following instructions for assigning terms to concepts:

- There are 2800 terms total and 6 categories (including other) that they should be classified into.
- Use a “1” to indicate that you consider a term to fall within a given concept and “0” to indicate that it does not.
- Try to use a narrow definition of the overriding concept; i.e, don’t include a term under that classification if there is another category that would fit it significantly better.
- However, if a single term can be reasonably considered to belong to more than one of concept, input a “1” for each respective concept.
- If you can imagine a concept which is a better fit for the term than the 5 provided, input a “1” under the ”Other” column.
- If you encounter any cases that you have difficulty classifying, please make note of it and reach out for guidance.

Additionally, research assistants were provided with the following definitions of the core concepts obtained from Webster Dictionary:

Biology

- a branch of knowledge that deals with living organisms and vital processes

Economy

- the structure or conditions of economic life in a country, area, or period

Executive

- one that exercises administrative or managerial control

Government

- the complex of political institutions, laws, and customs through which the function of governing is carried out

Sport

- physical activity engaged in for pleasure