Explainable Zero-Shot Topic Extraction Using a Common-Sense Knowledge Graph

Ismail Harrando
EURECOM, Sophia Antipolis, Biot, France

Raphaël Troncy
EURECOM, Sophia Antipolis, Biot, France

Abstract

Pre-trained word embeddings constitute an essential building block for many NLP systems and applications, notably when labeled data is scarce. However, since they compress word meanings into a fixed-dimensional representation, their use usually lack interpretability beyond a measure of similarity and linear analogies that do not always reflect real-world word relatedness, which can be important for many NLP applications. In this paper, we propose a model which extracts topics from text documents based on the common-sense knowledge available in ConceptNet [24] – a semantic concept graph that explicitly encodes real-world relations between words – and without any human supervision. When combining both ConceptNet's knowledge graph and graph embeddings, our approach outperforms other baselines in the zero-shot setting, while generating a human-understandable explanation for its predictions through the knowledge graph. We study the importance of some modeling choices and criteria for designing the model, and we demonstrate that it can be used to label data for a supervised classifier to achieve an even better performance without relying on any humanly-annotated training data. We publish the code of our approach at `https://github.com/D2KLab/ZeSTE` and we provide a user friendly demo at `https://zeste.tools.eurecom.fr/`.

2012 ACM Subject Classification Computing methodologies → Information extraction

Keywords and phrases Topic Extraction, Zero-Shot Classification, Explainable NLP, Knowledge Graph

Digital Object Identifier 10.4230/OASIcs.LDK.2021.17

Supplementary Material *Software (Source Code)*: `http://github.com/D2KLab/ZeSTE`
archived at `swh:1:dir:1d6fc42dda71a72d6869ff6a2ba46c209e82cf07`
InteractiveResource (Website): `https://zeste.tools.eurecom.fr/`

Funding This work has been partially supported by the French National Research Agency (ANR) within the ASRAEL (ANR-15-CE23-0018) and ANTRACT (ANR-17-CE38-0010) projects, and by the European Union's Horizon 2020 research and innovation program within the MeMAD (GA 780069) and SILKNOW (GA 769504) projects.

1 Introduction

Word2Vec [14], GloVe [16], BERT [5] along with its many variants are among the most cited works in NLP. They have demonstrated the possibility of creating generic, cross-task, context-free and contextualized word representations from big volumes of unlabeled text, which can be then used to improve the performance of numerous down-stream NLP tasks by bringing free "real world knowledge" about words meanings and usage, learned mostly through word co-occurrences statistics, thus cutting down the need for substantial amounts of labeled data. However, being compacted representations of word meanings, these embeddings do not offer much in terms of interpretation: we know that similar words tend to have similar representations (i.e. similar orientation in the embedding space), and that some analogies can be found by doing linear algebraic operations in the embedding space (such as the

3rd Conference on Language, Data and Knowledge (LDK 2021).
Editors: Dagmar Gromann, Gilles Sérasset, Thierry Declerck, John P. McCrae, Jorge Gracia, Julia Bosque-Gil, Fernando Bobillo, and Barbara Heinisch; Article No. 17; pp. 17:1–17:15
OpenAccess Series in Informatics
OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

now-famous $v_{King} - v_{Man} + v_{Woman} \approx v_{Queen}$). Both measures, however, fall short when evaluated systematically, as there is an entire literature about studying the limits of analogies and the biases that these word embeddings can encode depending on the corpora they have been trained on [4, 2, 15, 13].

In this paper, we consider the task of *topic categorization*, a sub-task of text classification where the goal is to label a textual document such as a news article or a video transcript, into one of multiple predefined *topics*, i.e. labels that are related to the topical content of the document. Common examples for news topics are *"Politics"*, *"Sports"* and *"Business"*. What is interesting about this task, compared to other text classification tasks such as *spam detection* or *sentiment analysis*, is that the content of the document to classify is *semantically related* to the labels themselves, providing an interesting case for zero-shot prediction setting. Zero-shot prediction, broadly defined, is the task of predicting the class for some input without having been exposed to any labeled data from that class.

To do so, we propose to leverage *ConceptNet*, a knowledge graph that aims to model common sense knowledge into a computer- and human-readable formalism. Coupled with its graph embeddings (ConceptNet Numberbatch[1]), we show that using this resource does not only achieve better empirical results on the task of zero-shot topic categorization, but also does so in an explainable fashion. With every word being a node in the knowledge graph, it is straightforward to justify the similarity between words in the document and its assigned label, which is not possible for other distributional word embeddings as they are built on the statistical aggregations of large volumes of textual data.

The remainder of this paper is structured as follows: we present some related work for text categorization emphasizing the methods that make use of external semantic knowledge (Section 2). We present our proposed method, named **ZeSTE** (**Ze**ro **S**hot **T**opic **E**xtraction) in Section 3. We empirically evaluate our approach for zero-shot topic categorization in Section 4 where we compare it to different baselines on multiple topic categorization benchmark datasets (including a non-English dataset). We also test our method against a few-shot setup and show how our approach can be combined with a supervised classifier to obtain competitive results on the studied datasets without relying on any annotated data. In Section 5, we describe a demo that we developed that enable users to provide their own set of labels and observe the explanations for the model predictions. Finally, we conclude and outline some potential future improvements in Section 6.

2 Related Work

Nearly all recent state-of-the-art Text Categorization models ([29, 3, 28, 25], to cite a few) rely on some form of Transformer-based architecture [27], pre-trained on large text corpora. While the task of using fully-unsupervised, non-parametric models for text categorization is yet to be explored to the best of our knowledge, there has been multiple efforts to incorporate common-sense knowledge as a basis for many artificial intelligence tasks, especially in a zero-shot setting where humans seem to be able to satisfactorily perform a new task by relying mostly on their common sense and prior knowledge accumulated from their interaction with the world.

In this paper, we propose to leverage ConceptNet [24], a multilingual semantic graph containing statements about common-sense knowledge. The nodes represent concepts (words and phrases, e.g. `/c/en/sport`, `/c/en/belief_system`, `/c/en/ideology`, `/c/fr/coup_d'_état`) from 78 languages, linked together by semantic relations such as `/r/IsA`, `/r/RelatedTo`,

[1] https://github.com/commonsense/conceptnet-numberbatch

/r/Synonym, /r/PartOf. The graph contains over 8 million nodes and 21 million edges, expressed in triplets such as (/c/en/president, /r/DefinedAs, /c/en/head_of_state). It was built by aggregating facts from the Open Mind Common Sense project [20], parsing Wiktionary[2], Multilingual WordNet [8], OpenCyc [7], as well as a subset of DBpedia, and designed to explicitly express facts about the real world and the usage of words and concepts that is necessary to understand natural language. Along with the graph, *ConceptNet Numberbatch* are multilingual pre-trained word (and concept) embeddings that are built on top of the ConceptNet knowledge graph. They are generated by computing the Positive Pointwise Mutual Information (PPMI) for the matrix representation of the graph, reducing its dimensionality, and then using "expanded retrofitting" [23] to make them more robust and linguistically representative by combining them with Word2Vec and GloVe embeddings. While the approach can be carried using other linguistic resources such as WordNet [8], we choose to use ConceptNet because it models word relations that are more relevant to the task of Topic Categorization such as /r/RelatedTo, which is the most present relation in the graph.

[6] is an early example of leveraging semantic knowledge to improve text categorization. It uses the relations in WordNet [8] to enhance the Bag of Word representation of documents by mapping the different words from a document into their entries in WordNet, and adding those as well as their hypernyms to the Bag of Words count. This, followed by a statistical χ^2 test to reduce the dimension of the feature vector, leads to a significant improvement over the simple bag-of-word model. [21] introduces *Graph of Words*, in which every document is represented by a graph of its terms, all connected with relations reflecting the co-occurrence information (terms appearing within a window of size w are joined by an edge). The authors propose a weighting scheme for the traditional TF-IDF model, where nodes are weighted based on some graph centrality measure (degree, closeness, PageRank), and edges are weighted with Word2Vec word embedding cosine similarity between their nodes. Incorporating both graph structure and distributional semantics from the embeddings to compute a weight for each term yields significantly better results on multiple text classification datasets.

[30] benchmark the task of zero-shot text classification, underlining the lack of work reported on this challenge in the NLP community in comparison to the field of computer vision. They distinguish two definitions of zero-shot text categorization: *Restrictive*, in which during a training phase, the classifier is allowed to see a subset of the data with the corresponding labels, but during inference, it is tested on a new subset of examples from the same dataset but not pertaining to any of the seen labels; *Wild*, where the classifier is not allowed to see any examples from the labeled data but can use Wikipedia's categories as a proxy dataset, for example. Our method fits into this second definition, although it does not require any training data. The authors compare some methods in both regimes (restrictive and wild) and they propose "Entail", a model based on BERT [5] and trained on the task of textual entailment evaluated on the Yahoo! Comprehensive Questions and Answers dataset.

[17] tackle the task of zero-shot text classification by projecting both the document and the label into an embedding space and using multiple architectures to measure the relatedness of the document and label embeddings. At test time, the classifier is able to ingest labels that were not seen during the training phase, but share the same embedding space with the labels already seen. A similar approach is followed by [22], in which both documents and labels are embedded into a shared cross-lingual semantic representations (CLESA) built upon Wikipedia as a multilingual corpus, and then the prediction is made by measuring the similarity between the two representations.

[2] https://en.wiktionary.org/wiki/Wiktionary:Main_Page

Finally, [31] propose a two-stage framework for zero-shot document categorization, combining 4 kinds of semantic knowledge: distributional word embeddings, class descriptions, class hierarchy, and the ConceptNet knowledge graph. In the first phase, a (coarse-grained) classifier is trained to decide whether the document at hand comes from a class that was seen during the training phase or not. This is done by training one ConvNet classifier [11] per label in the "seen" dataset, and setting a confidence threshold that, if none of the classifiers meets, the document is considered to be for the unseen labels. Secondly, a fine-grained classifier predicts the document final label. If the document is from a "seen" label, then the corresponding pretrained ConvNet classifier is picked. Otherwise, a zero-shot classifier which takes as input a representation of the document, the label, and their ConceptNet closeness, is trained on the seen labels but is expected to generalize to unseen ones as they share the same embedding space.

3 Approach

Our approach aims to perform topic categorization without relying on any in-domain labeled or unlabeled examples. Our underlying assumption is that words belonging to a certain topic are part of a vocabulary that is semantically related to its humanly-selected candidate label, e.g. a document about the topic of *"Sports"* will likely mention words that are semantically related to the word *Sport* itself, such as *team*, *ball*, and *score*. We use ConceptNet [24] to produce a list of candidate words related to the labels we are interested in. We generate a "topic neighborhood" for each topic label which contains all the semantically related concepts/nodes, and we then compute a score for each label based on the document content. Figure 1 illustrates our approach using a simple example.

3.1 Generating Topic Neighborhoods

To generate the topic neighborhoods for a given label, we query ConceptNet for nodes that are directly connected to the label node. Since the number of calls to the online API is capped at 120 queries/minute, we instead use the dump[3] of all ConceptNet v5.7 assertions, keeping only the English and French concepts for the English and French datasets, resulting in 3,323,321 (resp. 2,943,446) triplets, respectively. Although the assertions contain a finer granularity when it comes to referring to concepts, we only consider the root word for each concept to build the neighborhood. For example, the word "match" has multiple meanings: the tool to light a fire `/c/en/match/n/wn/artifact`, the event where two contenders meet to play `/c/en/match/n/wn/event`, and the concept of several things fitting together `/c/en/match/n/wn/cognition`. All these nodes (as well as others such as the verb form) will be mapped to the same term: "match". We also add (inverse) relations from the object to the subject for each triplet to ensure that every term in the graph has a neighborhood. The total number of unique triplets is 6,412,966, with 1,165,189 unique nodes for English (6.413.002 and 1.448.297 for French, respectively).

The topic neighborhood is created by querying every node that is N hops away from the label node. Every node is then given a score that is based on the cosine similarity between the label and the node computed using *ConceptNet Numberbatch* (ConceptNet's graph embeddings). This score represents the relevance of any term in the neighborhood to the main label, and would also allow us to refine the neighborhood and produce a score. In the

[3] `https://github.com/commonsense/conceptnet5/wiki/Downloads#assertions`

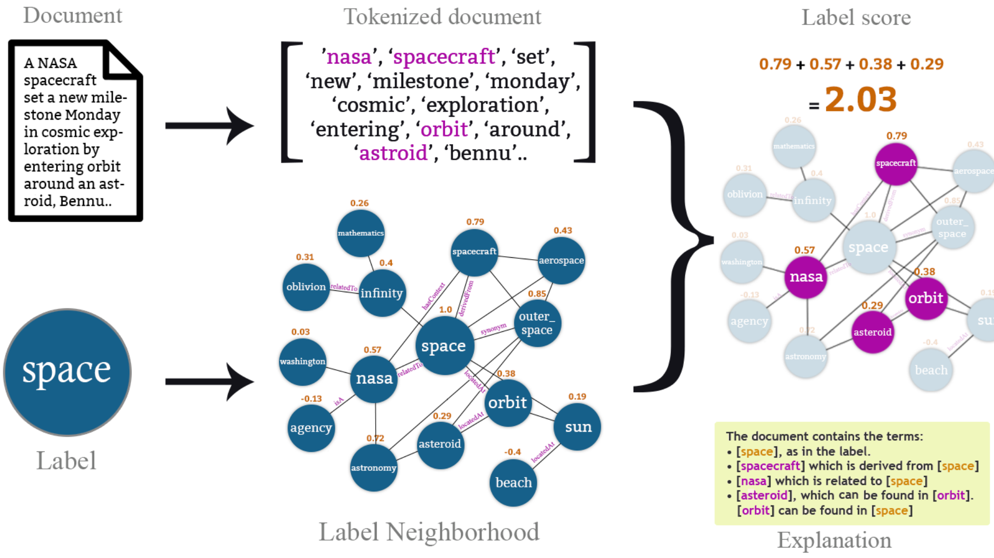

Document
A NASA spacecraft set a new milestone Monday in cosmic exploration by entering orbit around an astroid, Bennu..
Tokenized document
'nasa', 'spacecraft', 'set', 'new', 'milestone', 'monday', 'cosmic', 'exploration', 'entering', 'orbit', 'around', 'astroid', 'bennu'..
Label score
0.79 + 0.57 + 0.38 + 0.29
= 2.03
space
Label
0.26 mathematics
0.31 oblivion
relatedTo
0.4 infinity
0.79 spacecraft
0.43 aerospace
hasContext
derivedFrom
1.0 space
0.85 outer_space
synonym
0.03 washington
0.57 nasa
relatedTo
-0.13 agency
isA
0.72 astronomy
0.29 asteroid
locatedAt
0.38 orbit
0.19 sun
-0.4 beach
Label Neighborhood
The document contains the terms:
• [space], as in the label.
• [spacecraft] which is derived from [space]
• [nasa] which is related to [space]
• [asteroid], which can be found in [orbit]. [orbit] can be found in [space]
Explanation

Figure 1 Illustration of ZeSTE: given a document and a label, we start by pre-processing and tokenizing the document into a list of terms, and we generate the label neighborhood graph by querying ConceptNet (we omit some relation labels in the figure for clarity). Each node on the graph is associated with a score that corresponds to the cosine similarity between the graph embeddings of that node and the label node. We use the overlap between the document terms and the label neighborhood to generate a score for the label, as well as an explanation for the prediction. After doing so for all candidate labels, we pick the one with the highest score to associate to the document at hand.

case of a label which has multiple tokens (e.g. the topic "Arts, Culture, and Entertainment"), we just take the union of all word components' neighborhoods, weighted by the maximum similarity score if the same concept appear in the vicinity of multiple label components.

The higher N is, and the bigger the generated neighborhoods become. We thus propose multiple methods to vary the size of the neighborhood:

1. **Coverage**: we vary the number of hops N;
2. **Relation masking**: we consider subsets of all possible relations between words from the ConcepNet knowledge graph. More precisely, we consider three cases:
   a. The sole relation *RelatedTo* which is the most frequent one in the graph;
   b. The 10 semantic and lexical *similarity* relations only, i.e. *'DefinedAs', 'DerivedFrom', 'HasA', 'InstanceOf', 'IsA', 'PartOf', 'RelatedTo', 'SimilarTo', 'Synonym', 'Antonym'*;
   c. The whole set of 47 relations defined in ConceptNet.
3. **Filtering**: we filter out some nodes based on their similarity score:
   a. Threshold (*Thresh T*): we only keep nodes in the neighborhood if their similarity score to the label node is greater than a given threshold T.
   b. Hard Cut *(Top N)*: we only keep the top N nodes in the neighborhood ranked by their similarity score.
   c. Soft Cut *(Top P%)*: we only keep the top P% nodes in the neighborhood, ranked on their similarity score.

3.2 Scoring a Document

Once the neighborhood is generated, we can predict the document label by quantifying the overlap between the document content (as broken down to a list of tokens) and the label neighborhood nodes, which we denote in the following equations as $doc \cap LN(label)$. We consider the following scoring schemes:

1. **Counting**: assigning the document with the highest overlap count between its terms and the topic neighborhood.

$$count\_score(doc, label) = |doc \cap LN(label)| \tag{1}$$

2. **Distance**: factoring in the graph the distance between the term in the document and the label (number of nodes or path length between the token node and the label): the further a term is from the label vicinity, the lower is its contribution to the score.

$$distance\_score(doc, label) = \sum_{token \in doc \cap LN(label)} \frac{1}{min\_path\_length(token, label) + 1} \tag{2}$$

3. **Degree**: each node's score is computed using the number of incoming edges to it, reflecting its importance in the topic graph (we use $f(n) = log(1 + n_{edges})$ to amortize nodes with a very high degree).

$$degree\_score(doc, label) = \sum_{token \in doc \cap LN(label)} f(node\_degree(token)) \tag{3}$$

4. **Numberbatch similarity**: for each term in the document included in the label neighborhood, we increase the score by its similarity to the label embedding (we denote the Numberbatch concept embedding for word w by nb_w).

$$numberbatch\_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(nb_{token}, nb_{label}) \tag{4}$$

5. **Word Embedding similarity**: similar to the Numberbatch similarity, but we use pre-trained 300-dimensional GloVe [16] word embeddings instead to measure the word similarity (we denote the GloVe word embedding for word w by $glove_w$).

$$glove\_score(doc, label) = \sum_{token \in doc \cap LN(label)} sim(glove_{token}, glove_{label}) \tag{5}$$

We observe that in equations 4 and 5, multiple similarity measures and normalization options were considered, but the cosine similarity empirically showed the best results, so it has been used for the rest of the experiments. The model is thus the set of the neighborhood for each candidate label coupled with a scoring scheme. We discuss in Section 4.2 (Model Selection) how to empirically decide on the best filtering and scoring method that we then use in our experiments and our online demo.

3.3 Explainability

Given the label neighborhood, we can generate an explanation as to why a document has been given a specific label. This explanation can be generated in natural language or shown as the subgraph of ConceptNet that connects the label node and every word in the document that appears within its neighborhood, and hence counted towards its score3.1. We note that, although the "RelatedTo" edge does not offer much in term of explanation beyond semantic relatedness, its explicit presence in ConceptNet confirms this relatedness beyond any non-explicit measure (e.g. word embedding similarity). Since this graph is usually quite big, we can generate a more manageable summary by picking up the closest N terms to the label in the graph embedding space, as they constitute the nodes contributing most to the score of the document. We can show one path (for instance, the shortest) between each of the top term nodes and the label node. The paths can then be verbalized in natural language. For example, for the label `Sport`, and a document containing the word *Stadium*, a line from the explanation (i.e. a path on the explanation subgraph) would look like this (`r/RelatedTo` and `r/IsA` are two relations from ConceptNet):

> The document contains the word "Stadium", which is *related to* "Baseball". "Baseball" *is a* "Sport".

Another method of explaining the predictions of the model is to highlight the words (or n-grams) that contributed to the classification score in the document. Since every word that appear both in the document and the label neighborhood has a similarity score associated to it (e.g. the cosine similarity between the word and the label embedding), we can visually highlight the words that are relevant to the topic. These two explanation methods are further discussed in the Section 5.

4 Experiments

In this section, we first describe the datasets which have been used to evaluate our approach (Section 4.1). Next, we present experiments to select the best model (Section 4.2). We then detail the zero-shot baselines that we compare to our approach (Section 4.3) before discussing our results (Section 4.4). Finally, we show how our model can be used to bootstrap the training for supervised classifier to achieve significantly better results (Section 4.5).

4.1 Datasets

While the premise of our approach is the possibility to perform topic categorization in a zero-shot setting, we evaluate it on several datasets from the literature. We identify 4 different Topic Categorization datasets with different properties in terms of style (professional news sources or user-generated content), size, number of topics, topic distribution and document length. We also evaluate our model on a new dataset named AFP News, which provides interesting comparison grounds such as multilingualism (available in English and French), multi-topical documents and strong imbalance in topics distribution. Table 3 summarizes the characteristics of each of these 5 datasets.

- **20 Newsgroups** [12]: a collection of 18000 user-generated forum posts arranged into 20 groups seen as topics such as *"Baseball", "Space", "Cryptography", and "Middle East"*.
- **AFP News** [18]: a dataset containing 125K English and 26K French news articles issued by the French News Agency (*Agence France Presse*). The articles are tagged with one or more topics coming from IPTC NewsCode taxonomy[4]. We consider the first level of this taxonomy which corresponds to 17 top-level topics such as *"Art, Culture and Entertainment", "Environment", or "Lifestyle and Leisure"*. The label distribution is

[4] `http://cv.iptc.org/newscodes/subjectcode/`

highly unbalanced. Since the data on both the English and French documents come from the same source and have similar properties, we use this dataset to compare how well our method compare on two different languages.

- **AG News** [10]: a news dataset containing 127600 English news articles from various sources. Articles are fairly distributed among 4 categories: *"World", "Sports", "Business" and "Sci/Tech".*
- **BBC News** [9]: a news dataset from BBC containing 2225 English news articles classified in 5 categories: *"Politics", "Business", "Entertainment", "Sports" and "Tech".*
- **Yahoo! Answers Comprehensive Dataset** [26]: a dataset containing over 4 million questions (title and body) and their answers submitted by users, extracted from the Yahoo! Answers website. We construct the evaluation dataset following the procedure described in [30] to reproduce its setup for comparison: we select 10K questions from each of the top 10 categories on Yahoo! Answers. We split it into 2 categories. The first split contains the labels *"Health", "Family & Relationships", "Business & Finance", "Computers and Internet" and "Society and Culture"* whereas the second split contains the labels *"Entertainment & Music", "Sports", "Science & Mathematics", "Education & Reference", and "Politics & Government".* The ground-truth topic labels are assigned by users.

In order to determine the filtering criteria as discussed in Section 4.2 without relying on any further dataset-specific tuning, we use the BBC News dataset as a development set to select the optimal parameters for our model, under the hypothesis that the properties that work best for this dataset would work best for others as well. We verify post-hoc that this hypothesis holds empirically, i.e., the design choices decided using BBC News turn out to deliver the best results on the other datasets as well. The filtering criteria values that gave the best results for *Threshold*, *Hard Cut* and *Soft Cut* have empirically been set to $T = 0.0$, $N = 20000$, $P = 50\%$, respectively.

The 5 datasets have all been pre-processed using the same procedure: we lowercase the text, remove all non-alphabetical symbols and English (or French) stopwords. We then tokenize the strings using the space as separator and finally lemmatize the word using `WordNetLemmatizer`[5]. If the dataset has multiple textual contents (e.g. the Yahoo! Questions dataset consists of questions that are made of a title, a question body, and a set of answers), we concatenate them to form one "document". In the case of the AFP News dataset, each document can be tagged with one label, multiple labels, or no labels. We drop all non-tagged documents. To compute accuracy, we consider a prediction to be correct if it is among the document labels, and false otherwise. Finally, for the 20 Newsgroups dataset, we collapse the categories "comp.os.ms-windows.misc" and "comp.windows.x" into "windows", and "comp.sys.mac.hardware" and "comp.sys.ibm.pc.hardware" into "hardware", since they have very similar original labels. We do so for the baselines methods as well.

4.2 Model Selection

In this section, we evaluate some of the options regarding the neighborhood filtering and document scoring mentioned in Section 3. We use the *BBC News* dataset as a testbed for evaluating model selection. We report the results on the other datasets using the best parameters found at this stage. We first evaluate the different choices made to generate the label neighborhood as discussed in Section 3.1 and reported in Table 1.

[5] `http://www.nltk.org/api/nltk.stem.html?highlight=lemmatizer#module-nltk.stem.wordnet`

Table 1 Comparing the different filtering configurations on the BBC News dataset (performance expressed in Accuracy).

		Filtering method			
Relations	Depth	Keep All	Top50%	Top20K	Thresh
One	N = 1	55.4	54.5	55.4	55.4
	N = 2	69.0	65.8	64.8	66.2
	N = 3	81.0	81.3	83.5	81.3
Similarity	N = 1	60.8	57.5	60.8	60.8
	N = 2	70.3	66.9	66.2	68.0
	N = 3	77.9	81.9	83.4	81.9
All	N = 1	68.4	674	68.4	68.4
	N = 2	75.2	73.8	78.0	73.9
	N = 3	83.6	83.6	84.0	83.6

We observe that the most consistent way of improving the results is to use larger neighborhoods, as 3-hops neighborhoods systematically outperform the 1 and 2-hops ones. Our experiments show that going beyond $N = 3$ comes at the cost of increasing the computation time (mainly the computation of cosine similarity between the label and related nodes), while offering only very marginal improvement overall. The filtering method also impacts the performance but not as consistently (especially for $N = 3$). Finally, using all the relations generally yields better results than using only a subset of the relations, enough to justify the speed trade-off. It is also worth noting that using only the "r/RelatedTo" relation yields comparatively good results, which highlights the fact that "common-sense word relatedness" as expressed in ConceptNet is a strong signal for topic categorization.

For the scoring scheme, we evaluate the various methods mentioned in Section 3.2. The results are reported in Table 2.

Table 2 Evaluating the scoring schemes on BBC News (performance expressed in Accuracy).

Count	Distance	Degree	Numberbatch	GloVe
81.8	77.8	78.1	84.0	81.6

We see that using the ConceptNet Numberbatch embeddings gives the best result as they can condense the count, distance, degree of the nodes and the linguistic similarity with regard to the label into a measure of similarity in the embedding space. Accounting for term frequency (counting a word twice in the scoring if it appears twice in the document) in all of the scoring schemes did not translate to an improvement on the results. Accounting for n-grams, however, seems to slightly improve the results, but they require the availability of a corpus to mine such n-grams. Therefore, for the rest of our experiments, we do not account for n-grams. For the rest of our experiments, we keep the following configuration: *("All relations", $N = 3$, "Top20K", "Numberbatch scoring").* We use ConceptNet v5.7 and Numberbatch embeddings v19.08.

Table 3 Performance on five Topic Categorization datasets (Accuracy).

Dataset	BBC News	AG News	20 Newsgroups	AFP News (FR)	YQA-v0	YQA-v1
# topics	5	4	20	17	5	5
# docs	2225	127600	18000	125516	50000	50000
doc/topic std	54.3	22.4	56.7	13682.7	0.0	0.0
Avg.words/doc	390	40	122	242	43	44
EN	26.1	26.7	53.5	60.0	51.8	36.2
GWA	40.2	63.9	36.7	32.8	49.9	43.4
Entail [30]	71.1	64.0	45.8	61.8	52.0	49.3
ZeSTE	**84.0**	**72.0**	**63.0**	**80.9** (**78.2**)	**60.3**	**58.4**
Supervised	96.4	95.5	88.5		72.6	80.6
Method	[19]	[29]	[28]		[30]	

4.3 Baselines

We propose 3 baseline systems:

- *Entail*: this model is provided by HuggingFace[6] [30]. We use `bart-large-mnli` as our backend Transformer model which can also be tested at `https://huggingface.co/zero-shot/`.
- *GloVe Weighted Average* (GWA) inspired by [1]: we average the 300-d GloVe embeddings vectors for every word in the document, and use the cosine similarity between the document embedding and the GloVe label embedding as a score to classify the document. For multi-worded labels (e.g. "Middle East"), we use the average vector of all the label components as the label embedding.
- *Embedding Neighborhood* (EN): for each label, we select the 20k closest words in the embedding space. We score each document by adding up the cosine similarity between the GloVe embedding of every word in the document that appears in the "embedding neighborhood" and the GloVe embedding of the label. In other words, we substitute the explicit graph connections in ConceptNet with the closeness in the GloVe embedding space. This baseline reflects the ability of generic embeddings to encode the topicality of words based only on the similarity in the embedding space.

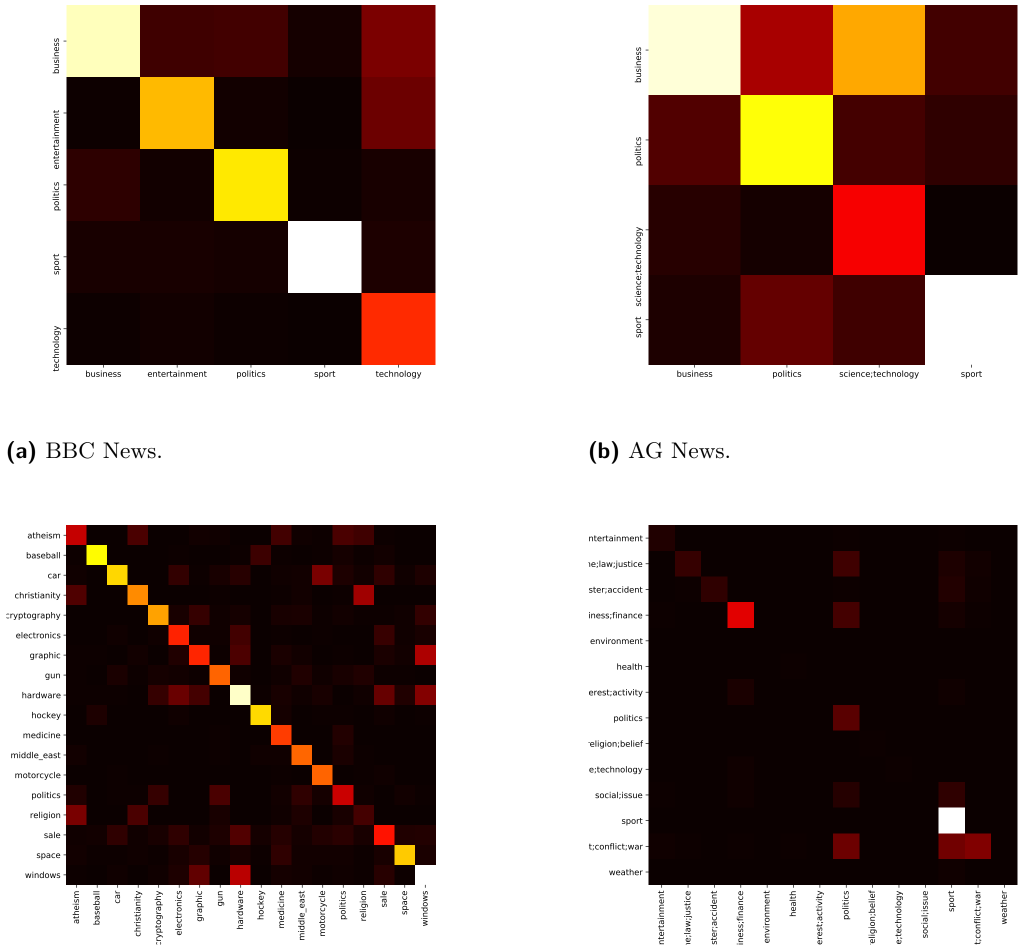
4.4 Zero-Shot Results

We provide the results obtained by evaluating our method against the baselines on the 5 datasets (BBC News, AG News, 20 Newsgroups, AFP News and YQA) in Table 3. Our method surpasses both GloVe baselines with a significant margin in accuracy on all datasets. `GWA` shows that the generic word embeddings poorly encode the topicality of words, as it is based solely on the similarity scores between the document content and the label world embedding. The low results with `EN` show that filtering based only on the embedding space (instead of the graph) is insufficient since the rarely-used words tend to clutter the embedding neighborhood. `ZeSTE` significantly outperforms `Entail`, despite the fact that the later relies on a large corpus pre-training and *textual entailment* task fine-tuning.

The confusion matrices for each datasets (Figure 2) indicate that our method performs more poorly on datasets where there is a lot of topical overlap between the different labels. For example, on 20 Newsgroups, "alt.atheism", "soc.religion.christian", "talk.religion.misc"

[6] We are using the implementation provided at `https://github.com/katanaml/sample-apps/tree/master/01`

have a lot of overlapping vocabulary, leading to most documents under “alt.atheism” to fall into either other options. If we collapse all three labels into one (e.g. “religion”), the performance improves from 63.0% to 68.9%. We also observe on the AFP News dataset that “politics” intersects with “unrest, conflict, war” and “business, finance”. The lack of a diameter pattern in AFP’s confusion matrix is due to the high imbalance in the labels, which hurts the precision of the model. It is also worth mentioning how the method works seamlessly for other languages, as demonstrated on the French AFP News dataset, which sees a slight drop of accuracy from 80.9% on English to 78.2% accuracy on French. This shows a great potential for multilingual applicability as ConceptNet supports 78 languages.



(a) BBC News.

(b) AG News.

(c) 20 NewsGroup.

(d) AFP News.

Figure 2 Confusion Matrices for the 4 news datasets.

Our method is clearly outperformed by the fully supervised methods. While the drop in performance is significant for some datasets, it is to be observed that the supervised methods not only rely on the availability of labeled training data, but usually also require expensive pre-training on more data. For instance, [29] use XLNet, an autoregressive Transformer that has been pre-trained on 120 GB of text. We consider that this absolute loss of accuracy performance is counter-balanced by the applicability in a zero-shot setting as well as the explainability of the model’s decision.

Finally, we note that the choice of the initial label can be critical for the functioning of this method. While we stayed true to the original labels in the experiments (with an exception for the label "World" that was replaced with "news, politics" in the AG News dataset), we are aware of the possibility of obtaining even better results by changing a label to a more fitting one or including more keywords into it.

4.5 Few-Shots Setup

For each dataset, we compare our model to a more realistic use-case. We create a 80-20 training/test split if one is not already provided, and we randomly sample n examples from each category to create a training set for our supervised classifier. Among the classifiers considered, we find uncased BERT (*BertForSequenceClassification*) to perform the best. We grow n in increments of 10 until we achieve an empirical accuracy score on the test set that surpasses our approach in the zero-shot setting. We report $N = n * |labels|$ the number of documents that need to be annotated in Table 4. We also observe that increasing the number of documents does not always improve the test set accuracy.

Table 4 The required number of documents needed to achieve zero-shot best performance.

Dataset	BBC News	AG News	20 Newsgroups	AFP News
N	300	240	2160	8500

4.6 Bootstrapping a Supervised Classifier

One of the potential usage of zero-shot classification is to provide "automatic labeling" for unlabeled documents to a traditional supervised classifier. In other words, we use ZeSTE to annotate a portion of each dataset, and we feed these annotated examples to a state-of-the-art text classifier.

We first define the confidence of the classification as the normalized score for each label, i.e. divided by the sum of all candidate labels scores. In Figure 3, which shows the error distribution with respect to the classification confidence, we see that it correlates well with whether the label is correct or not. Therefore, we can use it as a signal to pick samples to use to bootstrap our classifier. We train the same few-shots model from 4.5 on the best 60% examples of our training data, i.e. we drop 40% of the training examples on which ZeSTE is least confident. We report on the results in Table 5 (the results for ZeSTE row correspond to the performance on the test-set only, not the entire dataset as in Table 3). We can clearly see how the bootstrapping process helps the classifier achieving significantly better results on all tested datasets, all without requiring any human annotation. It is worth mentioning that for this application, the BERT-based classifier training was not thoroughly fine-tuned, which means that even better results can be achieved using the same automatic labeling setup.

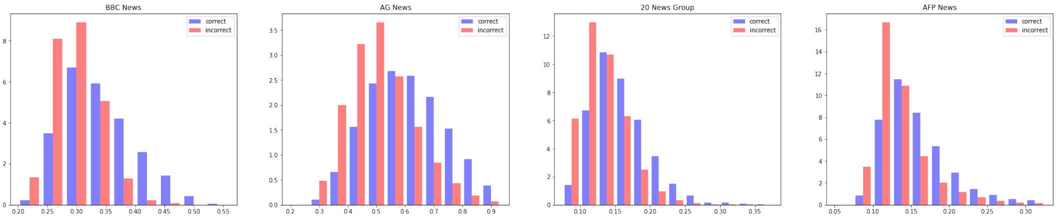


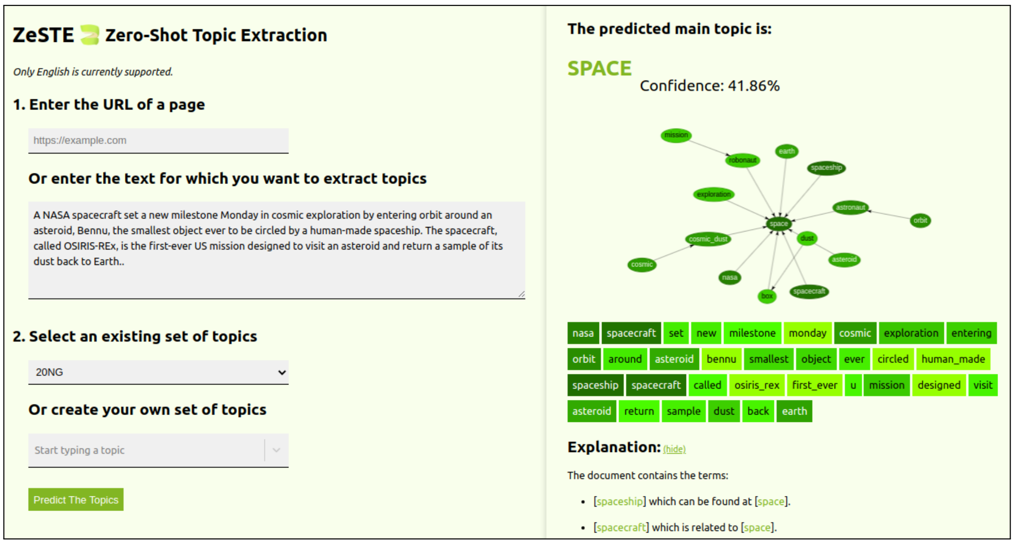
Figure 3 The prediction error distribution along the normalized confidence scores.

Table 5 The accuracy of ZeSTE and used as bootstrapped model (using the generated predictions as training data) on the test split of each dataset.

Dataset	BBC News	AG News	20 Newsgroups	AFP News
ZeSTE	80.6	71.0	61.6	73.8
ZeSTE + BERT	**94.3**	**84.2**	**70.1**	**83.0**

5 Online Demo

To demonstrate our method, we developed a web application which allows users to create their own topic classifier in real time. The user inputs the text to classify either by typing it into the designated textbox or by providing the URI of a web document that we scrape for extracting the content using Trafilatura[7]. The user is then prompted to either choose one of the pre-defined sets of labels (e.g. 20NG or IPTC used to evaluate the AFP dataset), or to provide her own set of label candidates. Once the user clicks on the "Predict the Topics" button, the server computes and caches the label neighborhood if it is the first time it encounters the label, otherwise it loads it from the cache for near real-time topic inference. Once the document is pre-processed and the label neighborhood generated, the server sends back its predictions (as confidence scores for each label candidate), and an explanation for each topic based on the common-sense connections between the document content and the label is provided (Figure 4, right panel). We only sample one path between document terms and the label, when in reality there could be many, in order to have a usable UI. In the future, we aim to depict the explanation as a subgraph of ConceptNet which shows all the relevant terms and their connections in the label neighborhood. We also highlight the relevant words in the input text (based on their score). While the demo works only for textual document written in English, we expect to support other languages in the future. The user interface makes use of the ZeSTE API which we also expose for others to be easily integrated.



ZeSTE Zero-Shot Topic Extraction

Only English is currently supported.

1. Enter the URL of a page

https://example.com

Or enter the text for which you want to extract topics

A NASA spacecraft set a new milestone Monday in cosmic exploration by entering orbit around an asteroid, Bennu, the smallest object ever to be circled by a human-made spaceship. The spacecraft, called OSIRIS-REx, is the first-ever US mission designed to visit an asteroid and return a sample of its dust back to Earth..

2. Select an existing set of topics

20NG

Or create your own set of topics

Start typing a topic

Predict The Topics

The predicted main topic is:

SPACE

Confidence: 41.86%

nasa spacecraft set new milestone monday cosmic exploration entering orbit around asteroid bennu smallest object ever circled human_made spaceship spacecraft called osiris_rex first_ever u mission designed visit asteroid return sample dust back earth

Explanation: (hide)

The document contains the terms:

- [spaceship] which can be found at [space].
- [spacecraft] which is related to [space].

Figure 4 ZeSTE's User Interface deployed at `https://zeste.tools.eurecom.fr/`.

[7] `https://pypi.org/project/trafilatura/`

6 Conclusion and Future Work

In this work, we present ZeSTE, a novel method for zero-shot topic categorization that achieves competitive performance for this task, outperforming solid baselines and previous works while not requiring any labeled data. Our method also provides explainable predictions using the common-sense knowledge contained in ConceptNet. We demonstrate that ZeSTE can help to bootstrap a supervised classifier, achieving high accuracy on all datasets without requiring human supervision. The code to reproduce our approach and replicate our results is available at `https://github.com/D2KLab/ZeSTE`.

As an extension to this work, we consider an adaptation of the approach to other NLP tasks such as multi-class topic categorization, query expansion and keyphrase extraction. To further improve the approach, an analysis on how to partition the topic neighborhoods and minimise overlap is also envisaged. Finally, studying how to automatically pick better topic labels based on measures such as Mutual Information and Graph Centrality is to follow.

References

1 Katherine Bailey and Sunny Chopra. Few-shot text classification with pre-trained word embeddings and a human in the loop. arXiv, 2018. `arXiv:1804.02063`.

2 Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in neural information processing systems*, pages 4349–4357, 2016.

3 Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal Sentence Encoder. arXiv, 2018. `arXiv:1803.11175`.

4 Dawn Chen, Joshua C Peterson, and Thomas L Griffiths. Evaluating vector-space models of analogy. arXiv, 2017. `arXiv:1705.04416`.

5 Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 4171–4186. Association for Computational Linguistics, 2019.

6 Zakaria Elberrichi, Abdelattif Rahmoun, and Mohamed Amine Bentaalah. Using WordNet for Text Categorization. *International Arab Journal of Information Technology (IAJIT)*, 5(1), 2008.

7 Charles Elkan and Russell Greiner. Building large knowledge-based systems: Representation and inference in the Cyc project. *Artificial Intelligence*, 61(1):41–52, 1993.

8 Ingo Feinerer and Kurt Hornik. *wordnet: WordNet Interface*, 2017. R package version 0.1-14. URL: `https://CRAN.R-project.org/package=wordnet`.

9 Derek Greene and Pádraig Cunningham. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *23rd International Conference on Machine learning (ICML)*, pages 377–384, 2006.

10 Antonio Gulli. *AG's corpus of news articles*, 2005. URL: `http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html`.

11 Yoon Kim. Convolutional neural networks for sentence classification. arXiv, 2014. `arXiv:1408.5882`.

12 Ken Lang. Newsweeder: Learning to filter netnews. In *12th International Conference on Machine Learning (ICML)*, pages 331–339, 1995.

13 Chandler May, Alex Wang, Shikha Bordia, Samuel R. Bowman, and Rachel Rudinger. On Measuring Social Biases in Sentence Encoders. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pages 622–628. Association for Computational Linguistics, 2019.

14 Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

15 Orestis Papakyriakopoulos, Simon Hegelich, Juan Carlos Medina Serrano, and Fabienne Marco. Bias in Word Embeddings. In *International Conference on Fairness, Accountability and Transparency (FAT)*, pages 446—457. Association for Computing Machinery, 2020.

16 Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

17 Pushpankar Kumar Pushp and Muktabh Mayank Srivastava. Train once, test anywhere: Zero-shot learning for text classification. arXiv, 2017. `arXiv:1712.05972`.

18 Charlotte Rudnik, Thibault Ehrhart, Olivier Ferret, Denis Teyssou, Raphaël Troncy, and Xavier Tannier. Searching News Articles Using an Event Knowledge Graph Leveraged by Wikidata. In *5th Wiki Workshop*, pages 1232–1239, 2019.

19 Vishal S Shirsat, Rajkumar S Jagdale, and Sachin N Deshmukh. Sentence level sentiment identification and calculation from news articles using machine learning techniques. In *Computing, Communication and Signal Processing*, pages 371–376. Springer, 2019.

20 Push Singh, Thomas Lin, Erik T Mueller, Grace Lim, Travell Perkins, and Wan Li Zhu. Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences On the Move to Meaningful Internet Systems*, pages 1223–1237, 2002.

21 Konstantinos Skianis, Fragkiskos Malliaros, and Michalis Vazirgiannis. Fusing document, collection and label graph-based representations with word embeddings for text classification. In *12th Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs)*, New Orleans, Louisiana, USA, 2018.

22 Yangqiu Song, Shyam Upadhyay, Haoruo Peng, Stephen Mayhew, and Dan Roth. Toward any-language zero-shot topic classification of textual documents. *Artificial Intelligence*, 274:133–150, 2019.

23 R. Speer and Joshua Chin. An Ensemble Method to Produce High-Quality Word Embeddings. arXiv, 2016. `arXiv:1604.01692`.

24 Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *31st AAAI Conference on Artificial Intelligence*, 2017.

25 Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to Fine-Tune BERT for Text Classification? arXiv, 2019. `arXiv:1905.05583`.

26 Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to rank answers on large online qa collections. In *46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 719–727, 2008.

27 Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. arXiv, 2017. `arXiv:1706.03762`.

28 Felix Wu, Tianyi Zhang, Amauri Holanda de Souza Jr, Christopher Fifty, Tao Yu, and Kilian Q Weinberger. Simplifying graph convolutional networks. arXiv, 2019. `arXiv:1902.07153`.

29 Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763, 2019.

30 Wenpeng Yin, Jamaal Hay, and Dan Roth. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. arXiv, 2019. `arXiv:1909.00161`.

31 Jingqing Zhang, Piyawat Lertvittayakumjorn, and Yike Guo. Integrating semantic knowledge to tackle zero-shot text classification. arXiv, 2019. `arXiv:1903.12626`.