Zero-Shot Taxonomy Mapping for Document Classification

Lorenzo Bongiovanni
LINKS foundation
Torino, Italy
lorenzo.bongiovanni@linksfoundation.com

Luca Bruno
LINKS foundation
Torino, Italy
luca.bruno@linksfoundation.com

Fabrizio Dominici
LINKS foundation
Torino, Italy
fabrizio.dominici@linksfoundation.com

Giuseppe Rizzo
LINKS foundation
Torino, Italy 1
giuseppe.rizzo@linksfoundation.com

ABSTRACT

Classification of documents according to a custom internal *hierarchical taxonomy* is a common problem for many organizations that deal with textual data. Approaches aimed to address this challenge are, for the vast majority, supervised methods, which have the advantage of producing good results on specific datasets, but the major drawbacks of requiring an entire corpus of annotated documents, and the resulting models are not directly applicable to a different taxonomy. In this paper, we aim to contribute to this important issue, by proposing a method to classify text according to a custom hierarchical taxonomy *entirely without* the need of labelled data. The idea is to first leverage the semantic information encoded into pre-trained Deep Language Models to assigned a prior relevance score for each label of the taxonomy using zero-shot, and secondly take advantage of the hierarchical structure to reinforce this prior belief. Experiments are conducted on three hierarchically annotated datasets: WebOfScience, DBpedia Extracts and Amazon Product Reviews, which are very diverse in the type of language adopted and have taxonomy depth of two and three levels. We first compare different zero-shot methods, and then we show that our hierarchy-aware approach substantially improves results across every dataset.

CCS CONCEPTS

• **Information systems** → **Document representation**; • **Computing methodologies** → **Information extraction**; *Unsupervised learning*; *Transfer learning*; • **Applied computing** → *Document metadata*;

KEYWORDS

Natural Language Processing, Zero-Shot Text Classification, Hierarchical Text Classification

SAC '23, March 27-March 31, 2023, Tallinn, Estonia

ACM ISBN 978-1-4503-9517-5/23/03…$15.00
https://doi.org/10.1145/3555776.3577653

ACM Reference Format:
Lorenzo Bongiovanni, Luca Bruno, Fabrizio Dominici, and Giuseppe Rizzo. 2023. Zero-Shot Taxonomy Mapping for Document Classification. In *The 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23), March 27-March 31, 2023, Tallinn, Estonia.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3555776.3577653

1 INTRODUCTION

Classifying documents according to a custom taxonomy, is a fairly common problem one will sooner or later face when working with a lot of documents. Some examples of real-world challenges related to this task are: automatic categorization of documents into a hierarchical structure, the possibility of performing some downstream statistical analysis on the newly formed structure, enhanced explainability of own document bases to justify decision making. Thus, this kind of problem, known as *hierarchical text classification* (HTC), has aroused widespread attraction in both the industry and the academia. However, the main challenge is that many times, especially in the industry setting, this task is not well suited to be approached in the standard supervised fashion. This is because often taxonomies are very prone to changes, either due to the nature of the taxonomies themselves or, for example, because they are being developed by trial and error. Under these circumstances, collecting and labelling a corpus of many thousands of documents, which is already very expensive and time consuming when done a single time, becomes impossible to be repeated every time the taxonomy changes.
Fortunately, since the introduction of Transformers [14], advancements in Deep Language Modeling (DLM) [2, 8, 15] have shown the ability of the latter to encode a good deal of general semantics inside their own weights. This, in return, opens the way for approaching many text classification tasks in a zero-shot fashion, allowing, in some cases, to avoid manual annotation altogether.

In this work, we developed a self-contained method to classify documents according to a custom hierarchical taxonomy without the use of any extra data or manual annotation. Our method can be summarized into three main steps: *first*, we introduce *Zero-shot Semantic Text Classification (Z-STC)*, that leverages Deep Language Models to generate a prior score for each label of the taxonomy, representing the degree of alignment between label semantics and document semantics. *Secondly*, for each label, the distribution of Z-STC scores is computed on a ground set of randomly crawled Wikipedia articles, in order to statistically determine a threshold

α that represents the value of Z-STC score for which the label is highly likely to correctly describe a document content. *Lastly*, a novel method, *Upwards Score Propagation (USP)*, is used to combine labels Z-STC scores and thresholds α, in order to propagate scores through the levels of the taxonomy.
The Zero-Shot Taxonomy Mapping code, with all the scripts to reproduce the results reported in this paper, is made available on github [1].

The rest of the paper is structured as follows: in Section 2, we highlight relevant related works. In Section 3, we provide a high-level overview of our methodology and we introduce the three datasets used for evaluation. In Section 4, we introduce the Zero-shot Semantic Text Classification method and we compare performance of several models. In Section 5, the Upward Score propagation mechanism is discussed in detail. Finally, brief conclusions are drawn in Section 6.

2 RELATED WORK

Many works study how to *leverage the hierarchical structure* of taxonomies in the context of Hierarchical Text Classification (HTC): the authors of [10] address sparsity of data by considering documents relative to the taxonomy node (local information), but also information relative to all nodes that connect the current node to the root of the taxonomy (global information), and finally classification is performed by combining these two sources of information based on a dynamically computed mixture weight. In [7], the hierarchical structure of the taxonomy is taken into account by training a different Deep Neural Network on each node of the taxonomy. In this way, each classifier is specialised only on a subset of topics. The authors of [5] build a Deep Learning approach to model both local and global information at each level of the taxonomy. First, a representation of the document and of the taxonomy is learnt, then an attention mechanism is used to model dependencies in a top-down fashion, while a recurrent neural network keeps memory of the sequence. Eventually, a final classifier to decides if a document should be labelled with a particular node. In [1], HTC is formulated as a Sequence-to-Sequence problem, where the input sequence is text and the output is the sequence of taxonomy nodes from the root to the appropriate document label. An LSTM based Encoder-Decoder is then trained to model this new problem.
From all these works, we learn the importance of modelling both local and global information when classifying according to hierarchical structure. However, our approach differs in that we do not want to rely on labelled data.
Unsupervised text classification has also been studied: in [6], a set of training sentences is automatically created using a list of hand-picked keywords for each category. Then, after handling false positives, the newly created sentences are used to train a Naive Bayes model for text classification. The authors of [3] model the task as a text similarity problem between documents projected onto a latent space using Latent Semantic Analysis (LSA), and a list of keywords assigned to the target category by leveraging WordNet and expert human annotation. Words from both documents and keywords are then replaced by Word Embeddings and the cosine similarity is computed between documents and categories. Both these approaches rely heavily on manual definition of a list of keywords to define the categories for classification, which we wish to avoid completely by leveraging the semantics of the category name itself.
In [13], the Deep Language Model SBERT [11] is used to encode documents into a semantic vector space. The authors, then, mine the five nearest neighbors for every datapoint, which yields a weakly supervised training set over which they fine-tune Siamese networks. At test time, they group the document embeddings produced by this weakly supervised model into as many clusters as there are categories, and assign each label to the most likely cluster. This work is aligned to ours in that it exploits the ability of Deep Language Models to encode the semantics of documents into a vector space. However, we argue that the semantics of the categories is not leveraged at all neither in the process of clustering nor in the training of the Siamese networks.

Finally, two works that are very well aligned with our own effort on *Zero-shot Text Classification (TC)* are [16] and [4], where Transformer based Deep Language Models are employed for zero-shot multi label classification of text. [16] proposes to deal with zero-shot TC as a textual entailment problem by converting each label into the hypothesis $\mathcal{I}$ = *"This document is about *label*"*. BERT is, then, fine-tuned on three Natural Language Inference (NLI) tasks, and used to decide if the hypothesis $\mathcal{I}$ entails the document, in which case the label gets assigned to it. Although this method is very much aligned with our idea of zero-shot TC, the downside is that there is a high degree of arbitrariness introduced by the choice of how the hypothesis $\mathcal{I}$ is formulated.
The authors of [4] re-formulate the typical text classification task in a more universal 0/1 problem, where BERT processes both text and label as input and it has to predict 1 if the label actually describes the text, 0 otherwise. They show that this new paradigm helps the transferability of the fine-tuned model. We argue that the complexity of this model grows linearly with the number of labels, as both text and labels have to be seen by the model at the same time, which can become a problem if one is trying to deal with a taxonomy that can easily have hundreds of labels.

3 METHODOLOGY

Our method combines several elements to be able to both understand how the document is semantically related to each label of the taxonomy, and to leverage the hierarchical structure of the latter in order to reinforce labels relevance score:

(1) perform *Zero-shot Semantic Text Classification (Z-STC)* (Section 4) a simple method that produces zero-shot state-of-the-art prior scores for each label of the taxonomy purely based on semantics. These scores represent the likelihood of a label to be relevant for the document in object. In this step, the hierarchical structure of the taxonomy is disregarded and the task is essentially standard zero-shot text classification. We compare multiple DLMs to find the one best suited for Z-STC, and we also compare with existing zero-shot text classification methods;

[1] https://github.com/bong-yo/TaxonomyZeroShooter

(2) determine a *Relevance Threshold* α (Section 5.2) specific for each label of the taxonomy. This threshold is automatically selected by the statistical distribution of prior Z-STC scores of each label over a set of fixed encyclopedic documents and it represents the minimum relevance score of a label for it to be considered relevant to a document with high confidence;
(3) apply the *Upwards Score Propagation (USP)* (Section 5.1) method that propagates confidence scores from the lowest level of the taxonomy up, leveraging prior Z-STC scores, Relevance Thresholds α and the hierarchical structure of the taxonomy.

We validate our approach by testing every step on three publicly available annotated datasets: first we verify that the Z-STC step is solid, by considering multiple Semantic Text Embedding (STE) models on the task of raw (not hierarchical) Zero-Shot Text Classification and by comparing our Z-STC approach with other two state of the art Zero-Shot approaches. After choosing the best performing model for Z-STC, we run the Relevance Threshold algorithm to statistically determine which value of similarity indicates high relevance of each label to documents. Finally, we apply the Upwards Score Propagation mechanism to bring everything together and include the taxonomy structure information into the classification task. We compare results obtained in this way with the ones obtained by simply performing raw Z-STC using the 'flatten' taxonomy, i.e., without Relevance Thresholding and the USP mechanism, and we show that the results are greatly improved on all layers affected of all the datasets considered.

3.1 Datasets

To validate the robustness of our method, we selected three Hierarchical Text Classification datasets, very diverse in content, language and type of labels. Here, documents are labelled according to one, and just one, branch of the taxonomy relative to the dataset, i.e., each document will be labelled with one label from *Level 1*, one label from *Level 2*, and so on till the lowest level of the taxonomy.

3.1.1 ***Web Of Science (WoS).*** [2] is a collection of almost 50K research abstracts gathered and labelled in [7]. The language here is the one used in scientific papers and labels are technical keywords representing areas of research. The taxonomy has a two-level hierarchy, with 7 and 134 labels respectively.

3.1.2 ***DBpedia Extract.*** [3] contains 340K articles from Wikipedia, labelled according to DBpedia taxonomy. Here, there are 3 levels, with 9, 70 and 219 labels respectively. The language and categories are neutral, clean and informative in the style of Wikipedia.

3.1.3 ***Amazon Product Reviews.*** [4] This dataset contains about 50K products reviews, each one classified according to a three-level hierarchical taxonomy, provided by Amazon, with 6, 64 and 510 labels respectively. The language here differs greatly from the other two datasets in that it mostly contains very informal texts of customers reviewing some products they bought online through Amazon.

[2]https://huggingface.co/datasets/web_of_science
[3]https://www.kaggle.com/danofer/DBpedia-classes
[4]https://www.kaggle.com/kashnitsky/hierarchical-text-classification

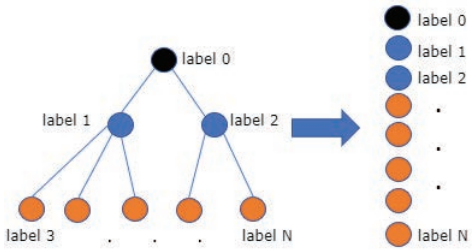

Figure 1: Flattening of the taxonomy from the original hierarchical structure to a list of labels

4 ZERO-SHOT SEMANTIC TEXT CLASSIFICATION

4.1 Semantic Text Embedding (STE)

Leveraging the knowledge latent in the weights of Transformer-based Deep Language Models allowed to train models [11] to perform extremely well on many different *Semantic Text Embedding (STE)* tasks, like Question Answering (QA), Passage Retrieval, Semantic Text Similarity (STS), Paraphrase, Natural Language Inference (NLI), or a mix of those. The ability of these models to capture the semantics of text allows, in general, to quantify how much a label is relevant to a document purely based on the semantics of both, exactly as a human would do, without the need for specific supervised training on annotated examples.
We are going to leverage these properties of the STE models to generate *prior relevance score* for each label in our taxonomy. Because these models have been pre-trained on a huge amount of textual data but never on the text classification task that we are going to use them on, we are going to refer to this process as *zero-shot*.

4.2 Zero-Shot Semantic Text Classification (Z-STC)

Zero-shot Semantic Text Classification (Z-STC) is the process of leveraging an STE-based text encoder Ψ to *separately* map the text of a document d and a taxonomy label l into the same semantic vector space, where a *prior relevance scores* $p(l)$ can be assigned simply by looking at the cosine similarity between the two:

$$p_d(l) = S_c(\Psi_D(d),\ \Psi_L(l)),$$
$$S_c(A, B) = \frac{A \cdot B}{|A||B|} \tag{1}$$

where the closer $p(l)$ is to 1 the more confidently D can be assigned to the label l. Here Ψ_D and Ψ_L represent the different use of the STE model when encoding document text and labels respectively, as discussed in Section 4.3.

At this stage the hierarchical structure of the taxonomy is flattened and every label l is considered as independent, as shown in Figure 1. The task becomes then a standard text classification problem, with the additional challenge of solving it in a zero-shot way without the help of annotated data. However, in contrast to other zero-shot approaches mentioned in Section 2, our Z-STC method encodes labels and documents *separately*. This allows the

complexity to be linear in the number of labels helping scaling to large number of labels (more detail on this in Section 4.4).

It is important to note that Z-STC differs from all other STE tasks mentioned in Section 4.1, in that it aims to embed labels, i.e. *short keywords*, in the same semantic space it embeds fully contextualized paragraphs of text. Standard STE models, instead, are solely trained on fully contextualized text. For this reason, in Section 4.5, we will take particular care on evaluating the performance of existing STE models on the Z-STC task we defined.

4.3 Entropy-Based Sentence Selection (ESS)

4.3.1 Document encoding. Generating a good embedding for text of arbitrary length poses, in general, two challenges: 1) Transformers based Language Models have, by design, a maximum input length, and 2) not all sentences of a document are useful for classification. Here, we address both challenges by dividing the full text into its sentences, encoding each one separately, and then allow the encoder Ψ_D to focus more on the most informative ones. More formally, given the text d of a generic document, and a pre-trained STE model Φ, we want to find the best encoding function $\Psi_D(d, \Phi)$ such that Equation (1) is optimal with respect to the set of given labels. As an example, consider we want to classify the following document:

'We are happy we can share our results here. The paper is about Natural Language Processing for Text Classification. Submitted to Journal A. All rights reserved'

according to the set of labels: *Quantum Physics*, *Medicine*, *Computer Science*.
We clearly want our encoder Ψ_D to be able to focus on the second sentence, while mostly ignoring the other three. In general, we consider a sentence s to be *informative* if it aligns very well with only few labels and not with the other ones. Fortunately, *entropy* is the perfect quantity to look at to know if a distribution is peaked around some values or spread over all possible values. In particular, given a sentence s we can compute its *normalized entropy*:

$$e_s = \frac{\sum_l \tilde{p}(l) \log_2(\tilde{p}(l))}{\log_2(N)}, \qquad \tilde{p}(l) = \frac{|S_c(\Phi(l), \Phi(s))|}{\sum_l |S_c(\Phi(l), \Phi(s))|} \tag{2}$$

where N is the number of labels, $\log_2(N)$ is the maximum possible entropy and $\tilde{p}(l)$ is the normalized version of $p(l)$ in Equation (1), to fit with the definition of entropy, where both labels and sentences are encoded with a straightforward application of a pre-trained STE model Φ. As defined in Equation (2), the sentence entropy has its minimum ($e_s = 0$) when only one label has maximum probability, while it has its maximum ($e_s = 1$) when all labels have same probability, which makes it perfectly aligned with our definition of *informativeness* of a sentence.

We can now construct the embedding of the document d by taking the average of the embeddings of each sentence weighted on their own entropy:

$$\Psi_D(d) = \frac{\sum_{s \in d}(1 - e_s)\Phi(s)}{\sum_{s \in d}(1 - e_s)} \tag{3}$$

Model	F1 (micro) on WoS
naive encoding Φ	0.596
ESS encoding Ψ	**0.606**

Table 1: Effect of the ESS encoding on the Zero-Shot classification of the first layer of Web Of Science (WoS) dataset. The text encoder adopted is *mpnet-all*, introduced in Section 4.5.

where $\Phi(s)$ represents the STE model encoding the sentence s. According to Equation (3), more informative sentences, i.e. with lower associated entropy e_s, will weight more in the resulting document embedding.

4.3.2 Label Encoding. Labels are short and informative keywords representing a class (e.g. 'Biochemistry'), therefore there is no need to use ESS and we simply use the STE model Φ to encode the label l as it is:

$$\Psi_L(l) = \Phi(l). \tag{4}$$

The positive effect of ESS is reported in Tab.1, in the context of Zero-Shot classification of the fist level of the Web Of Science dataset, described in 3.1.2, where the STE Encoder adopted is *mpnet-all*, discussed in detail in 4.5. In the first row, we encode the document by simply passing the whole text to the encoder (naive encoding $\Phi(d)$), then, in the second row, we apply the ESS method described above (ESS encoding $\Psi(d)$).

4.4 Similarity Complexity

According to the Z-STC paradigm introduced above, the model encodes *separately* the N labels of the taxonomy and the text of the M documents to classify. Only at this point the cosine similarity label-document is computed. The complexity of our approach, in terms of model forward passes, is, then, $O(N+M)$, as for every new document only the document text requires to be encoded by the model. Obviously, we still need to compute the cosine similarities between document and label embeddings which is a $O(N \times M)$ matrix multiplication operation whose computational time is, however, negligible compared to a deep model forward pass and, therefore, we can safely ignore it when talking about method complexity.

In contrast, both the other state-of-the-art models for zero-shot text classification [4, 16], need every single label to be shown to the model together with the text of every single document for prediction, resulting in $O(N \times M)$ complexity. This difference in complexity becomes particularly important since we are dealing with taxonomies that can easily have hundreds, if not thousands, of labels, in which case prediction time becomes quickly prohibitive. This can be seen in Table 3 where the number of documents processed per second is almost two orders of magnitude higher for method that have complexity $O(N + M)$.

4.5 STE Models Comparison

Our definition of Z-STC requires using a Semantic Text Embedder (STE) to produce embeddings for labels, which are typically keywords formed by one or, at most, few words. However, all available STE models are mostly trained to capture the semantics of

	Model	F1 (macro)							
		WoS		DBpedia			Amazon		
		Lev. 1	*Lev. 2*	*Lev. 1*	*Lev. 2*	*Lev. 3*	*Lev. 1*	*Lev. 2*	*Lev. 3*
1	TARS	0.366	0.221	0.265	0.166	0.493	0.295	0.135	0.098
2	BART-MNLI	0.453	0.342	0.269	0.196	0.520	0.382	0.155	0.112
3	mpnet-paraphrase (Z-STC)	0.573	0.401	0.271	**0.349**	0.621	**0.593**	0.243	0.172
4	mpnet-multi-qa (Z-STC)	0.484	0.374	0.267	0.332	**0.646**	0.580	**0.263**	**0.202**
5	bert-msmarco (Z-STC)	0.335	0.349	0.284	0.267	0.555	0.521	0.198	0.180
6	roberta-all-large (Z-STC)	0.528	0.456	0.314	0.290	0.568	0.484	0.228	0.164
7	**mpnet-all (Z-STC)**	**0.596**	**0.462**	**0.317**	0.326	0.628	0.547	0.256	0.173

Table 2: Z-STC performance of several top STE models (row 3-7) against two established SOTA zero-shot models *TARS* and *BART-MNLI* (row 1-2), on three datasets.

Model	Size (MB)	doc/sec (avg.)	doc/sec (min.)	scaling (N docs M labels)
TARS	**418**	5.2	1.2	O(N x M)
BART-MNLI	777	4.4	0.8	O(N x M)
mpnet-paraphrase (Z-STC)	**418**	88	67	O(N + M)
mpnet-multi-qa (Z-STC)	**418**	82	67	O(N + M)
bert-msmarco (Z-STC)	**418**	93	73	O(N + M)
roberta-all-large (Z-STC)	1355	60	52	O(N + M)
mpnet-all (Z-STC)	**418**	**107**	**91**	O(N + M)

Table 3: Models comparison in terms of size (in MB), average and slowest documents processed per second on a V100 GPU, and scaling with number N of documents and M of labels .

context-rich text, therefore *we cannot simply refer to the reported performance of the models to select the best STE model to use for our Z-STC method*. On the contrary, we need to investigate their performance when it comes to encode the semantics of short keywords compared to context-rich documents (Ψ_L and Ψ_D respectively in Equation 1).

We proceed by selecting a handful of the top performing Deep Language Models specialised on Semantic Text Embedding tasks[5], compare them on our datasets relatively to the Z-STC task, and pick the best one to bring forward. We start from the top five STE performing models for comparison, to cover, at least in part, the variability of STE tasks and Deep Language Model training paradigms:
mpnet-paraphrase: MPNet [12] is a model pre-trained using a Masked and Permuted Language Modelling approach. This particular MPNet model has been fine-tuned for Semantic Text Embedding (STE) to be able to detect if a sentence is paraphrasing another[6].
mpnet-multi-qa: MPNet based model specialised on Semantic Search and Question Answering. It has been tuned on multiple datasets[7].

[5]For this we refer to Hugging Face's database of models evaluated for their quality to embedded sentences and to embedded search queries & paragraphs: https://www.sbert.net/docs/pretrained_models.html
[6]https://huggingface.co/sentence-transformers/paraphrase-mpnet-base-v2
[7]https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1

bert-msmarco: BERT-based model, also specialised on Question Answering[8], and trained on the MSMARCO dataset [9].
roberta-all-large: BERT-based model, is trained with the same general STE intent as the previous *all-mpnet-base-v2* model, and on the same dataset[9].
mpnet-all: all-round MPNet based model, fine-tuned for many use-cases over a large and diverse STE dataset of over 1 billion examples[10].

All the STE models described above are compared in Table 2, on the three datasets described in Section 3.1, where every document is assigned to the labels, one for each level of the taxonomy, with the highest relevance score, computed with the Z-STC method. We will also compare Z-STC performance of two existing methods, i.e. **TARS** (Task-Aware Representation of Sentences) [4] and **BART-MNLI** [16], that have different Zero-shot paradigms. Looking at the results in Table 2-3 we can conclude:

(1) *mpnet-all (Z-STC)* is the fastest model and outperforms all other STE models on most of the datasets. In particular it outperforms *roberta-all-large (Z-STC)*, fine-tuned on the exact same data, which has almost four times the number of parameters;

[8]https://huggingface.co/sentence-transformers/msmarco-bert-base-d
[9]https://huggingface.co/sentence-transformers/all-roberta-large-v1
[10]https://huggingface.co/sentence-transformers/all-mpnet-base-v2

(2) MPNet paradigm seems to be, in general, more robust than BERT when it comes to Z-STC, as it achieve consistently higher results on the three datasets considered;
(3) compared to the zero-shot paradigms of [4] and [16], our Z-STC approach seems to perform better overall, on the datasets considered, and be much faster, in good agreement with the complexity analysis mentioned in 4.4.

In light of these results, we are inclined to say that, out of the STE models considered, *mpnet-all (Z-STC)* is the best suited for the task of Zero-Shot Semantic Text Classification, since it achieves best or near-best performance across every taxonomy level of every dataset.

5 LEVERAGING TAXONOMY HIERARCHY

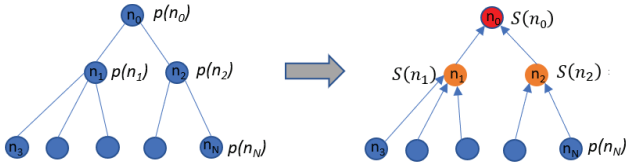

Figure 2: *Upwards Score Propagation (USP)* method. On the *left*, each node n is assigned a score $p(n)$ computed via Z-STC, and purely based on its own semantic. On the *right*, the score $\mathcal{S}_{USP}$, representing the re-calibration of confidence in p based on the structure of the taxonomy, gets propagated upwards according to Equation (5)

Using Zero-Shot Semantic Text Classification (Z-STC), as described in the previous section, we are able to generate a *prior relevance score* (Equation1) for each label of the taxonomy. Such score, however, expresses the likelihood of a label being relevant to a document only based on semantics, but not yet considering the labels position inside the hierarchy of the taxonomy. It is clear that the structure of the taxonomy contains a great deal of information that has to be taken into consideration when assigning relevance scores to labels. In particular, we will exploit the following paradigm:

*If a label is **relevant** to a document, then **also its parent label** is.*

In this section, we are going to describe the *Upwards Score Propagation (UPS)* method which updates prior relevance scores of labels into *posterior scores* by propagating confidence upwards through hierarchy of the taxonomy.

5.1 Upwards Score Propagation (USP)

We define the score function $\mathcal{S}_{USP}(l)$, which represents the *posterior relevance score* of the label l. $\mathcal{S}_{USP}$ depends on the prior confidence score $p(l)$, computed by Z-STC, on a label-specific threshold parameter α_l, discussed in the next section, that represents the scale of semantic similarity for the label l, and on the posterior scores of the children of the label l. Formally, if the label l has N children, one can define the recursive expression for every child c_i, with $i \in [1, N]$:

$$\mathcal{S}^{(i)}_{USP}(l) = \begin{cases} \mathcal{S}^{(i-1)}_l & \mathcal{S}_{c_i} \leq \mathcal{S}^{(i-1)}_l \\ \mathcal{S}^{(i-1)}_l \cdot e^{\left(\mathcal{S}_{c_i} - \mathcal{S}^{(i-1)}_l\right)} & \mathcal{S}^{(i-1)}_l \leq \mathcal{S}_{c_i} \leq \alpha_c \\ \mathcal{S}_{c_i} & \mathcal{S}_{c_i} \geq \mathcal{S}^{(i-1)}_l,\ \alpha_c \end{cases} \tag{5}$$

$$\mathcal{S}^{(0)}_l = max(0,\ p(l))$$
$$\mathcal{S}_{USP}(l) = \mathcal{S}^{(N)}_l$$

where we set $\mathcal{S}_{USP}(x) \equiv \mathcal{S}_x$ for readability. The final posterior relevance score for the label l, $\mathcal{S}_{USP}(l) = \mathcal{S}^{(N)}_l$, is consolidated after all the N children c have been taken into consideration. The initial value of every label l is simply its prior score $\mathcal{S}^{(0)}_l = max(0,\ p(l))$, where to ensure convergence of $\mathcal{S}_{USP}$, negative values of $p(l)$ are mapped into 0. This is justified by the fact that semantic similarity is captured by values of $p(l)$ close to 1, while dissimilarity is expressed by fluctuations around the value $p(l) = 0$, as it also shown by the shape of the distribution of labels similarity over unrelated texts (blue histogram in Figure 3). With this definition, the posterior scores $\mathcal{S}_{USP}$ remain naturally inside the interval $[0, 1]$ as

$$\begin{aligned} &\widehat{p}(l) = max(0,\ p(l)) \in [0, 1] \\ &\mathcal{S}_{USP}(l) = \widehat{p}(l) \cdot e^{\left(\mathcal{S}_{USP} - \widehat{p}(l)\right)} \leq \widehat{p}(l) \cdot e^{(1-\widehat{p}(l))} \leq 1 \end{aligned} \tag{6}$$

which allows the Upwards Score Propagation function to be applied recursively to any layer of the taxonomy.

According to Equation 5, the posterior relevance score $\mathcal{S}_{USP}(l)$ of the label l:

- remains the same, if the posterior score of the child c is lower than l's score;
- gets boosted by e^{Δ} if the posterior score of the child is greater than l's score. Here $\Delta \equiv \mathcal{S}_{c_i} - \mathcal{S}^{(i-1)}_l$ is the difference between the posterior score of the child and the score of the label;
- gets replaced entirely by the child posterior score, if that is greater than the child's Relevance Threshold α_c.

In this way we are allowing the USP function to propagate the information of a strongly relevant child label c to its parent label l. Moreover, particularly relevant children, i.e. those whose relevance score is above the Relevance Threshold α_c, can propagate their entire relevance score to the parent, respecting the intuition that if a child label is relevant for a document, then also its parent label is. A graphic representation of the USP process is given in Figure 2. It has to be noted that **USP does not affect the score of labels in the lowest taxonomy level**, since they do not have children labels to receive score propagation from.

5.2 Relevance Threshold α

The Relevance Threshold α has a central role for the function $\mathcal{S}_{USP}$ in Equation (5), as it represents the minimum value for which the score of a child label gets completely propagated to its parent label. The value of α can be interpreted as the minimum relevance

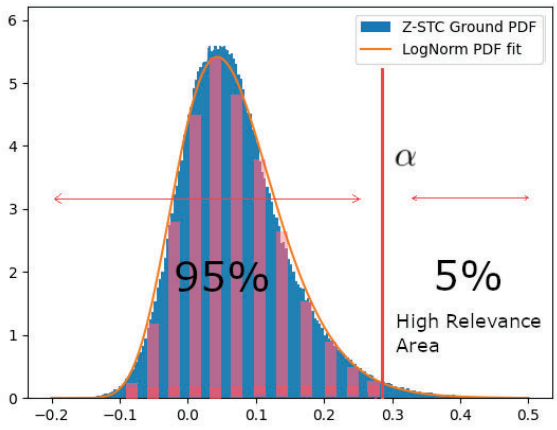

Figure 3: Ground Distribution of label relevance scores over 1000 randomly crawled Wikipedia articles (blue histogram) and its fit with a Log-Normal distribution (yellow line). The Relevance Threshold α is determined as the value that is higher than 95% of the Ground Distribution.

score of a label for which is *highly likely* that it is a correct label for a certain document.

5.2.1 Definition of statistical relevance for α. In statistics, it is common to refer to a value as *highly significant* when it *strongly deviates* from a given *Ground Distribution*. In our case, we consider the Ground Distribution of a label l to be the distribution of its scores $p_{GD}(l)$ over a set of *irrelevant* documents, and conclude that the label l is relevant to a new document d if its posterior relevance score $\mathcal{S}_{USP}(l)$ is statistically higher compared to its Ground Distribution.

The point of the the Ground Distribution is to be computed over a set of documents that is unrelated with the labels of the taxonomy we are using, for this reason we obtain the GD of irrelevant documents by computing the relevance scores $p_{GD}(l)$ of label l with over 1000 randomly selected Wikipedia articles (also included in the shared github repository). As customary for statistical relevance, we set α_l such that it is higher than 95% (2σ for Gaussian distribution) of the Ground Distribution, as shown in Figure 3. Any value $p_d(l) > \alpha_l$ indicates, therefore, that the label l is highly relevant for a given document d.

PDF	WoS	DBPedia	Amazon
Normal	-29.6	-39.5	-27.1
Gumbel	-10.7	-64.0	-14.1
Log-Normal	**-48.0**	**-77.4**	**- 45.3**

Table 4: Bayesian Information Criterion (BIC) to select the best out of three PDFs: Normal, Gumbel and Log-Normal. BIC is computed separately for each label and then averaged over the taxonomy of each dataset. Lower values of BIC indicate better model.

5.2.2 Bayesian Information Criterion (BIC) for model selection. It is important to choose the most appropriate Probability Distribution Function (PDF) to model the Ground Distribution of irrelevant Wikipedia articles, so that the 95% Relevance Threshold can be computed with accuracy for each label. The Ground Distribution (Figure 3) has a shape that resembles Gaussian but with an asymmetry towards the positive values. Therefore, we chose three possible candidates belonging to the family of exponential distributions: Gaussian, Gumbel and Log-Normal:

$$\begin{aligned}
\mathcal{N}(x;\mu,\sigma) &= \frac{1}{\sigma\sqrt{2\pi}} e^{-1/2(x-\mu)^2/\sigma^2} \\
\mathcal{G}(x;\beta,m) &= \frac{1}{\beta} e^{-\left((x-m)/\beta + e^{-(x-m)/\beta}\right)} \\
\mathcal{L}(x;\sigma,\theta,m) &= \frac{1}{(x-\theta)\sigma\sqrt{(2\pi)}} e^{-\ln((x-\theta)/m)^2/(2\sigma^2)}
\end{aligned} \tag{7}$$

We select the best fitting PDF by evaluating the **Bayesian Information Criterion (BIC)**:

$$\text{BIC} = k\ln(n) - 2\ln(\hat{L}) \tag{8}$$

for each of the three distributions in Equation (7) with respect to the Ground Distribution. The BIC is a widely adopted criterion for model selection, it favours the likelihood of the candidate distribution, given the data, while penalizing its complexity. *Lower BIC* indicates, usually, a *more suitable model*. The parameters in Equation (8) are:

- $\hat{L}$: the maximized value of the likelihood function of the candidate model $\mathcal{M}$, i.e. $\hat{L} = p(x \mid \widehat{\theta}, \mathcal{M})$, where $\widehat{\theta}$ are the parameter values that maximize the likelihood function;
- x: the observed data;
- n: the number of data points;
- k: the number of free parameters of the candidate model $\mathcal{M}$.

We compute the BIC of the three candidate PDFs separately for each label, then we average the results over the taxonomy of each dataset and report the values in Table 4. Log-Normal PDF shows a superior fit across every dataset and we, therefore, chose it as our modelling function. Finally, we are able to find the Relevance Threshold α_l for a label l by:

(1) fit $\mathcal{L}_l(x;\sigma,\theta,m)$ to find the value of the parameters $\hat{\sigma},\hat{\theta},\hat{m}$ that maximize its likelihood given the Ground Distribution;
(2) compute the Log-Normal Cumulative Distribution Function (CDF): $C_l(y) = \int_{-\inf}^{y} \mathcal{L}(x;\hat{\sigma},\hat{\theta},\hat{m})dx$, which represents the area up to y of $\mathcal{L}(x)$;
(3) set $\alpha_l = C_l(0.95)$, i.e., the value of similarity with l which is higher than 95% of the similarity of l with irrelevant documents.

These three steps are completely automated, given the set of fixed Wikipedia documents, and can be applied straight away to any custom taxonomy. Moreover, it is worth to notice that this operation is computed only once when the custom taxonomy is passed to the Zero-Shot Taxonomy Mapping module, and it does not depend on the number of documents to classify, therefore it is a constant computational factor that does not affect the overall complexity scaling of the module.

Model	F1 (macro)							
	WoS		DBpedia			Amazon		
	Lev. 1	*Lev. 2*	*Lev. 1*	*Lev. 2*	*Lev. 3*	*Lev. 1*	*Lev. 2*	*Lev. 3*
mpnet-all (Z-STC)	0.596	0.462	0.317	0.326	0.628	0.547	0.256	0.173
mpnet-all (Z-STC) + USP	**0.741**	0.462	**0.759**	**0.656**	0.628	**0.712**	**0.348**	0.173

Table 5: Performance of USP mechanism compared with 'raw' Z-STC on the task of hierarchical text classification. Both methods use the model mpnet-all to perform Z-STC. Performance on the last layer of each dataset remains unchanged since USP does not affect the last layer.

5.3 USP Performance

Finally, we proceed to evaluate the overall *Upwards Score Propagation* mechanism discussed so far, on the three annotated dataset. To summarize, the steps involved are the following:

(1) run *mpnet-all (Z-STC)* model to compute prior relevance scores for every label on every document;
(2) compute the Relevance Threshold α_l for every label in the taxonomy;
(3) apply USP to get posterior relevance scores;
(4) select the label with the highest posterior relevance score for each level of the taxonomy, and compare it with the one manually assigned by the annotators.

Results in Table 5 clearly show that the USP, **substantially improves performance across every datasets and affected taxonomy layer**, i.e. every layer except the last one. This is expected due to the fact that the labels in the lowest level of the taxonomy do not have further children labels, therefore they cannot benefit from the Upward Score Propagation method.

6 CONCLUSIONS

In this paper, we have discussed a novel self-contained method that is able to classify documents according to a hierarchical taxonomy without the need of any annotated data. Our experiments on three datasets have shown that Zero-Shot Semantic Text Classification (Z-STC) is able to produce state-of-the-art zero-shot prior relevance scores for labels, after which, Upwards Score Propagation (USP) greatly improves performance everywhere by leveraging the structure of the taxonomy.

ACKNOWLEDGMENTS

This work was partially funded by the European Union through the STARLIGHT project (H2020-SU-AI-2020), grant agreement n. 61273304, and APPRAISE project (H2020-SU-SEC-2020), grant agreement n. 101021981

REFERENCES

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *CoRR* abs/1802.02611 (2018). arXiv:1802.02611 http://arxiv.org/abs/1802.02611

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[3] Zied Haj-Yahia, Adrien Sieg, and Léa A. Deleris. 2019. Towards Unsupervised Text Classification Leveraging Experts and Word Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 371–379. https://doi.org/10.18653/v1/P19-1036

[4] Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task-Aware Representation of Sentences for Generic Text Classification. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 3202–3213. https://doi.org/10.18653/v1/2020.coling-main.285

[5] Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical Multi-Label Text Classification: An Attention-Based Recurrent Network Approach. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 1051–1060. https://doi.org/10.1145/3357384.3357885

[6] Youngjoong Ko and Jungyun Seo. 2000. Automatic Text Categorization by Unsupervised Learning *(COLING '00)*. Association for Computational Linguistics, USA, 453–459. https://doi.org/10.3115/990820.990886

[7] Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. HDLTex: Hierarchical Deep Learning for Text Classification. *CoRR* abs/1709.08267 (2017). arXiv:1709.08267 http://arxiv.org/abs/1709.08267

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *CoRR* abs/1907.11692 (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[9] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *CoRR* abs/1611.09268 (2016). arXiv:1611.09268 http://arxiv.org/abs/1611.09268

[10] Heung-Seon Oh, Yoonjung Choi, and Sung-Hyon Myaeng. 2011. Text Classification for a Large-Scale Taxonomy Using Dynamically Mixed Local and Global Models for a Node. In *Advances in Information Retrieval*, Paul Clough, Colum Foley, Cathal Gurrin, Gareth J. F. Jones, Wessel Kraaij, Hyowon Lee, and Vanessa Mudoch (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 7–18.

[11] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *CoRR* abs/1908.10084 (2019). arXiv:1908.10084 http://arxiv.org/abs/1908.10084

[12] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. MPNet: Masked and Permuted Pre-training for Language Understanding. *CoRR* abs/2004.09297 (2020). arXiv:2004.09297 https://arxiv.org/abs/2004.09297

[13] Dominik Stammbach and Elliott Ash. 2021. DocSCAN: Unsupervised Text Classification via Learning from Neighbors. *CoRR* abs/2105.04024 (2021). arXiv:2105.04024 https://arxiv.org/abs/2105.04024

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *CoRR* abs/1706.03762 (2017). arXiv:1706.03762 http://arxiv.org/abs/1706.03762

[15] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *CoRR* abs/1906.08237 (2019). arXiv:1906.08237 http://arxiv.org/abs/1906.08237

[16] Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. *CoRR* abs/1909.00161 (2019). arXiv:1909.00161 http://arxiv.org/abs/1909.00161