Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs

RICCARDO CANTINI, FABRIZIO MAROZZO, GIOVANNI BRUNO and PAOLO TRUNFIO,
University of Calabria

The growing use of microblogging platforms is generating a huge amount of posts that need effective methods to be classified and searched. In Twitter and other social media platforms, hashtags are exploited by users to facilitate the search, categorization and spread of posts. Choosing the appropriate hashtags for a post is not always easy for users, and therefore posts are often published without hashtags or with hashtags not well defined. To deal with this issue, we propose a new model, called HASHET (*HAshtag recommendation using Sentence-to-Hashtag Embedding Translation*), aimed at suggesting a relevant set of hashtags for a given post. HASHET is based on two independent latent spaces for embedding the text of a post and the hashtags it contains. A mapping process based on a multilayer perceptron is then used for learning a translation from the semantic features of the text to the latent representation of its hashtags. We evaluated the effectiveness of two language representation models for sentence embedding and tested different search strategies for semantic expansion, finding out that the combined use of BERT (*Bidirectional Encoder Representation from Transformer*) and a global expansion strategy leads to the best recommendation results. HASHET has been evaluated on two real-world case studies related to the 2016 United States presidential election and COVID-19 pandemic. The results reveal the effectiveness of HASHET in predicting one or more correct hashtags, with an average F-score up to 0.82 and a recommendation hit-rate up to 0.92. Our approach has been compared to the most relevant techniques used in the literature (*generative models*, *unsupervised models* and *attention-based supervised models*) by achieving up to 15% improvement in F-score for the hashtag recommendation task and 9% for the topic discovery task.

Additional Key Words and Phrases: Deep Neural Networks, Hashtag Recommendation, Sentence Embedding, Word Embedding, Social Media.

ACM Reference Format:
Riccardo Cantini, Fabrizio Marozzo, Giovanni Bruno and Paolo Trunfio, 2021. Learning sentence-to-hashtags semantic mapping for hashtag recommendation on microblogs. *ACM Trans. Knowl. Discov. Data.* V, N, Article A (May 2021), 26 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Social media platforms have become part of everyday life, allowing the interconnection of people around the world. Their intensive use leads to the generation of a huge amount of data, which hides a high exploitable intrinsic value. This data is well suited to a broad set of applications aimed at extracting relevant information about users behavior, interests and activities. One of the most widely used data sources comes from microblogging services such as Twitter, Facebook and Instagram. Microblogging is a form of small content publishing in a social network service, visible to everyone or only to people in the same community. This type of publication generates a large amount of posts which leads to the need for effective data categorization and search. To address this problem, posts often include one or more hashtags. A hashtag consists of a charac-

Author's addresses: R. Cantini and F. Marozzo and G. Bruno and P. Trunfio, DIMES, University of Calabria, Rende (CS), Italy. Email:{rcantini, fmarozzo, gbruno, trunfio}@dimes.unical.it

ter string preceded by the '#' symbol, and is used to organize posts according to content and context, in order to facilitate the search and spread of topics trends and create communities with similar interests. Choosing the appropriate hashtags for a post is not always easy for users, and therefore posts are often published without hashtags or with hashtags not well defined, which hinders the quality of search results [Godin et al. 2013]. In addition, the informal writing style and the length constraints make it difficult to analyze posts using traditional Natural Language Processing (NLP) methods. The success of deep learning, attention mechanisms and transformer architectures [Vaswani et al. 2017] in several NLP tasks has accelerated research in many fields related to social media analysis [Li et al. 2019]. Specifically, to deal with the hashtag recommendation task, recent works often rely on the construction of a common multi-modal embedding space in which data from multiple modalities (i.e., sentences and hashtags) could be projected. By inspecting this space, relative distances can be measured in order to find the most relevant matching sentence-hashtag pairs. [Kumar et al. 2019] followed this kind of approach proposing a Zero Shot Learning (ZSL) architecture based on a joint embedding model, where hashtags are projected in the embedding space of the sentences through an end-to-end learning process.

In this paper we propose a new model, called HASHET (*HAshtag recommendation using Sentence-to-Hashtag Embedding Translation*), which follows a different approach compared to the main related techniques. Instead of relying on a single multimodal joint embedding space, we reformulated this task as a translation between two independent embedding spaces: the semantic space of sentences and the latent space of hashtags. The former is obtained by using a pre-trained sentence embedding model, such as Universal Sentence Encoder (*GUSE*) [Cer et al. 2018] or Bidirectional Encoder Representations form Transformers (*BERT*) [Devlin et al. 2018], which are very effective in capturing semantic and syntactic features of microblog texts. The latter comes from the training of a Word2Vec model [Mikolov et al. 2013] based on a Continuous Bag of Words (CBOW) architecture, aimed at discovering contextual relationships between words and hashtags. Moreover, we inverted the direction of projection with respect to the most recent deep embedding models, learning a semantic mapping from the latent representation of a sentence to the embedding space of its hashtags.

Similarly to the attention-based models, we exploit a semantic representation of a microblog generated by a transformer-based encoder. The key difference is in how we use this representation to recommend hashtags. Neural-based solutions frame the recommendation task as a multi-class classification problem [Mahajan et al. 2018; Li et al. 2019; Gong et al. 2018], using a softmax activation and minimizing the cross-entropy loss. Differently, in HASHET, we translate the latent representation of a post into a target vector lying in the words/hashtags embedding space. Then, the top-k nearest hashtags are found and enriched using semantic expansion, a process based on semantic similarity in the hashtags embedding space. The obtained output is composed of semantically similar hashtags, reflecting the semantic relationships learned among hashtags and the underlying topic-based clustering structure. This inspection process exploits the locality in the words/hashtags embedding space, which introduces a marked improvement in predicting hashtags with respect to other techniques.

We evaluated the effectiveness of HASHET over two real-world case studies related to the 2016 United States presidential election and COVID-19 pandemic, achieving very promising results. In particular, HASHET - by jointly using BERT and a global expansion strategy - achieved an average F-score up to 0.82 and a hit-rate up to 0.92 for hashtag recommendation and an accuracy of 95% for topic discovery. Furthermore, experimental results show that HASHET significantly outperforms different competitive state-of-art methods (generative models, unsupervised models and attention-based supervised models) by achieving up to 15% improvement in F-score for the hashtag rec-

ommendation task and 9% for the topic discovery task. For the purpose of using the code of our method and allowing the reproducibility of the experiments, an open-source version of HASHET is available at https://github.com/scalabunical/HASHET.

The remainder of the paper is organized as follows. Section 2 presents the embedding techniques used in the proposed model. Section 3 discusses related work. Section 4 describes the HASHET model. Section 5 presents the case studies. Section 6 presents an analysis on the applicability of HASHET for real-time hashtag recommendation and Section 7 concludes the paper.

2. EMBEDDING TECHNIQUES

The HASHET model is based on a translation between two independent embedding spaces: i) the semantic space of sentences; ii) the latent space of hashtags. For what concerns the first embedding space (S_{emb}) we compared two of the most used state-of-art solutions for sentence encoding, published by Google, described in the following.

— *Google Universal Sentence Encoder* (*GUSE*) [Cer et al. 2018]. It consists in a deep sentence embedding model with two available implementations: one makes use of the transformer architecture [Vaswani et al. 2017], while the other is formulated as a deep averaging network (DAN) [Iyyer et al. 2015]. In this work, the first solution has been chosen for generating the latent representation of a given sentence. It is pre-trained on a variety of web sources and Stanford Natural Language Inference (SNLI) corpus [Bowman et al. 2015]. The encoding model is designed by using multi-task learning, according to which a single encoder is used for multiple downstream tasks, which are: a Skip-Thought like unsupervised task [Kiros et al. 2015], a conversational input-response task [Henderson et al. 2017], and a classification task for supervised learning. The latest transformer-based large version available on Tensorflow-Hub[1] has been used. Starting from a lowercase PTB tokenized string, it computes context aware representations of the input words, taking into account both their ordering and identity. These representations are then converted to a single 512-dimensional sentence encoding vector, computed as their element-wise sum.

— *Bidirectional Encoder Representations from Transformers* (*BERT*) [Devlin et al. 2018]. It is based on a multi-layer bidirectional Transformer [Vaswani et al. 2017], pre-trained on two unsupervised tasks, Masked Language Modeling (MLM) and Next Sentence Prediction (NSP), using a large crossdomain corpus. Unlike *OpenAI GPT* [Radford et al. 2018], which uses a unidirectional (left-to-right) language model or *ELMo* [Peters et al. 2018], which uses a shallow concatenation of independently trained left-to-right and right-to-left language models, BERT is deeply bidirectional. In fact, the use of MLM objective enables the representation to fuse the left and right contexts, allowing the pre-training of a deep bidirectional language representation model. BERT outperformed many task-specific architectures, advancing the state of the art in a wide range of Natural Language Processing tasks, such as textual entailment, text classification and question answering. In this work, we used the *bert-base-uncased* implementation from Huggingface[2], characterized by 12 Transformer blocks, a hidden dimensionality of 768, 12 attention heads and 110M parameters, exploiting the hidden representation of the *CLS* token as sentence embedding.

The second embedding space (W_{emb}) comes from the training of a Word2Vec model [Mikolov et al. 2013] based on a Continuous Bag of Words (CBOW) approach, aimed at learning a dense vector representation of the words in a given set of documents. The embedding process is based on semantic and syntactic similarity and both statistical

[1] https://tfhub.dev/google/universal-sentence-encoder-large/5
[2] https://huggingface.co/bert-base-uncased

and co-occurrence relationships with other words. Word2Vec is one of the most popular techniques to train a word embedding model using shallow neural networks. The training of such a model leads to the definition of a multidimensional latent space that reflects the semantic distribution of the words in the corpus. Words are represented uniquely as latent vectors and will be closer if recognized as semantically more similar, through notions such as cosine similarity. There are essentially two approaches to obtain an embedding with Word2Vec:

— Continuous Bag-of-Words (CBOW): given a fixed number of context words, this model tries to predict the word related to this context by distributing the probability on all the input terms with a single softmax output layer.
— Skip-Gram: starting from an input word, this model tries to predict the context, generating many probability distributions in the softmax output layer for how many context words are considered.

Since the HASHET model relies on the semantic mapping between the two aforementioned embedding spaces, it can be applied in the presence of social media posts in different languages as well as multilingual posts, which is a desirable property, as microblogging platforms are widespread across different cultures and geographies. In particular, the latent space of hashtags W_{emb} is language-independent, as it comes from a CBOW Word2Vec model trained from scratch on the given corpus of posts. For what concerns the pre-trained language representation models used for sentence embedding in the S_{emb} space, both present a multi-lingual version. In particular, the *Google Multilingual Universal Sentence Encoder* [Pires et al. 2019] embeds text from 16 languages into a single semantic space, while *Multilingual BERT* (mBERT[3]) covers 104 spoken languages from around the world.

3. RELATED WORK

In recent years, Natural Language Processing (NLP) has been attracting more and more interest by the scientific community. With the fast growing of microblog services, several NLP techniques have been developed for learning the representation of microblog posts and recommending pertinent hashtags. Existing techniques can be grouped into three main categories:

— *Generative models*. [Godin et al. 2013] proposed a method for suggesting the top hashtags for a given post. They exploited Latent Dirichlet Allocation for finding out the underlying topic distribution, used for recommending general hashtags. [Gong et al. 2018] proposed a generative model for recommending hashtags in multimodal microblog posts that combines textual and visual information. [She and Chen 2014] proposed a supervised topic model-based solution for hashtag recommendation on Twitter (TOMOHA). They treated hashtags as labels of topics, developing a supervised topic model for discovering relationships among words, hashtags and topics of tweets. Then, by inferring the probability that a hashtag will be contained in a new tweet, the k most probable ones are recommended.
— *Unsupervised models*. [Pang et al. 2015] investigated methods from the perspective of similarity diffusion, proposing a clustering-based method that exploits similarity cascades (SCs). SCs are a series of sub-graphs generated by truncating a similarity graph with a set of thresholds, where maximal cliques are used to capture topics. Topics are then identified through a process of similarity diffusion. [Ben-Lhachemi and Nfaoui 2018] proposed a hashtag recommendation methodology based on the embedded representation of Twitter microblog posts. They performed the following steps: i)

[3] https://github.com/google-research/bert/blob/master/multilingual.md

a given tweet is represented as the weighted average of its word embeddings; ii) latent representations of tweets are clustered according to their syntactic and semantic similarity using a density-based approach; iii) top-k hashtags are found by computing the similarity between the entered tweet and the centroids of the obtained clusters. [Huang et al. 2015] proposed a high utility pattern clustering (HUPC) framework over microblog streams. Starting from a group of representative patterns from the microblog stream, patterns that perform better in describing topics are grouped into clusters. In this way the proposed framework can detect coherent and new emerging topics simultaneously. [Otsuka et al. 2016] proposed a hashtag recommendation system for Twitter data streams, based on a novel ranking scheme, called Hashtag Frequency-Inverse Hashtag Ubiquity (HF-IHU), which is a variation of TF-IDF that considers hashtag relevancy and microblog data sparseness.

— *Attention-based supervised models*. In recent years, attention-based models proved to be very effective in a wide range of NLP tasks including summarization of sentences [Rush et al. 2015], or text entailment [Rocktäschel et al. 2015]. The basic idea behind the attention mechanism is to allow the model to focus on the relevant parts of the input sequence as needed. This goal is accomplished by determining a weight for each position that indicates the amount of attention that should be paid to it [Bahdanau et al. 2014; Luong et al. 2015]. The first contribution comes from [Bahdanau et al. 2014], who used an attention-based neural machine translation (NMT) approach to jointly translate and align words. Their model differs from a standard encoder-decoder model, as the input sentence is encoded into a sequence of vectors, weighted through the attention mechanism in order to generate the translation. [Lu et al. 2016] proposed a novel co-attention model for Visual Question Answering (VQA) that jointly reasons about image and question attention. [Feng et al. 2019] proposed a context-attention based Long Short-Term Memory network (CA-LSTM) for modeling a sequence of microblogging posts and classifying the related sentiment. Many researches have been carried out also in the hashtag recommendation field. [Gong et al. 2018] proposed a novel architecture based on convolutional neural networks enhanced with an attention mechanism for incorporating the trigger words. The authors formulated the hashtag recommendation task as a multi-class classification problem. They adopted an attention mechanism to scan input microblogs and select trigger words, which are combined with the whole microblog to perform the recommendation task. [Li et al. 2019] used an attention based neural network to learn the representation of a microblog post. Specifically, they proposed a novel Topical Co-Attention Network (TCAN) that models content attention and topic attention simultaneously. [Kumar et al. 2019] compared the performances of various deep learning architectures, such as recurrent neural networks or transformer-based architectures. They evaluated various state-of-art Zero Shot Learning methods like Convex combination of Semantic Embedding (ConSE), Embarrassingly Simple ZSL (ESZSL) and a Deep Embedding Model for ZSL (DEM-ZSL), based on a joint embedding space in which either tweets or hashtags are represented.

The HASHET model proposed in this paper effectively exploits the state-of-art techniques and recent deep learning architectures for natural language processing, such as embedding models and transformer networks, but follows a different semi-supervised approach:

— Instead of relying on a single multi-modal joint embedding space, we used two independent embedding spaces: the semantic space where sentences are embedded, and the latent space of hashtags.
— For obtaining the embedded representation of a given post, we compared two different transformer-based pre-trained encoders, in particular GUSE and BERT. The

use of these sentence embedding solutions, which exploit a self-attention mechanism [Vaswani et al. 2017], leads to a better semantic representativeness with respect to previous state-of-art techniques, which represent posts as the weighted average of their word embeddings. The embedding space of the hashtags, instead, has been learned by training a Word2Vec model using a Continuous Bag of Words architecture.
— We reformulated the hashtag recommendation task as a translation between the aforementioned embedding spaces, by learning a mapping from the embedded sentences to their latent hashtags. Moreover, these hashtags are jointly represented by a single vector, called target, which can be seen as a summarizing concept about them.
— We inverted the direction of the projection with respect to the most recent deep embedding models, by learning a semantic mapping from the latent representation of a sentence to the embedding space of its hashtags.
— Since semantically similar hashtags lie close together in the embedding space, our recommendation process exploits a concept of locality which relies on the semantic relationships within this space and the underlying topic-based clustering structure.
— We proposed two different strategies (local and global n-nhe) for semantic expansion, a process that allows an enrichment of the set of recommended hashtags, based on semantic similarity in the hashtag embedding space.

In order to evaluate the accuracy of HASHET, we carried out an extensive comparison with the most relevant techniques used in the literature. Among the aforementioned state-of-art models, the following have been selected: i) a *generative model* based on LDA and Gibbs sampling [Godin et al. 2013]; ii) two *unsupervised models*: HF-IHU, a frequency-based model exploiting a variation of TF-IDF ranking scheme [Otsuka et al. 2016], and a density-based model which exploits Word2Vec and DBSCAN algorithms [Ben-Lhachemi and Nfaoui 2018]; iii) three *attention-based supervised models*: TCAN, a neural model based on a topical co-attention network [Li et al. 2019], a Bi-directional LSTM model enhanced with global general-attention [Luong et al. 2015], and a fine-tuned BERT classifier [Devlin et al. 2018]. HASHET achieved very promising results, outperforming the other techniques in the hashtag recommendation and topic discovery tasks (for further details, see Section 5.1.3).

4. PROPOSED MODEL

The HASHET model is based on the embedded representation of a post in the semantic space S_{emb} and its projection in the latent space of its hashtags W_{emb}. The projection is performed by learning a translation between these independent spaces, through a semantic mapping based on a multilayer perceptron. As shown in Figure 1, the execution flow of HASHET consists of two main steps:

(1) Semantic mapping model creation and training.
(2) Latent space inspection and semantic expansion.

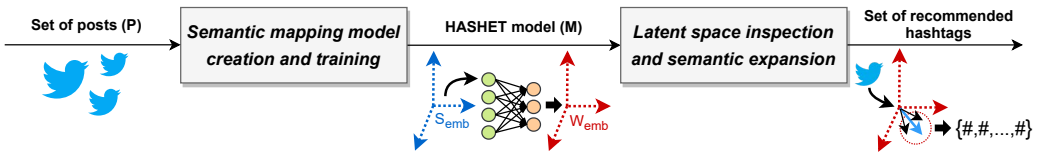


Fig. 1: Execution flow of HASHET, composed of two steps: 1) Creation and training of the semantic mapping model; 2) inspection of the hashtag latent space through semantic expansion for hashtag recommendation.

A formal description of each step is provided in the following subsections. For the reader's convenience, Table I reports the meaning of the main symbols used throughout the sections.

Symbol	Meaning
P	Corpus of posts.
E	The pre-trained encoder model (*GUSE* or *BERT*) exploited for sentence embedding.
S_{emb}	Latent space of sentence embedding. Dimensionality is 512 for *GUSE* and 768 for *BERT*.
$W2V$	The words/hashtags embedding model, based on CBOW Word2Vec.
W_{emb}	150-dimensional latent space of word embedding.
MLP	The mapper $S_{emb} \rightarrow W_{emb}$, based on a Multi-layer Perceptron.
SM	The semantic mapping model, obtained by stacking the mapper (*MLP*) on top of the encoder *E*.
$\mathcal{M}$	The HASHET model, defined as $\langle W2V, SM \rangle$.
$S_{emb}(p)$	Embedded representation of a post p in S_{emb}.
$W_{emb}(w)$	150-dimensional representation of a word w in W_{emb}.
$H(p)$	Set of the hashtags of the post p.
$target(p)$	Arithmetic mean of all the $W_{emb}(h), \forall\ h \in H(p)$.
$h^*(p)$	Projection of $S_{emb}(p)$ in the latent space W_{emb}. It is the output of SM given p as input.
$N^k(h)$	Ordered set of k nearest hashtags of h in W_{emb}.
$T^{k,n}(p)$	Set of top-k recommended hashtags for a post p expanded according to the factor n.

Table I: Meaning of the main symbols used.

4.1. Semantic mapping model creation and training

HASHET is based on the hidden relationships between the sentences and words/hashtags embedding spaces, S_{emb} and W_{emb}, learned by a semantic mapping process based on a multilayer perceptron. The main workflow of this step is shown in Figure 2 and described by Algorithm 1.

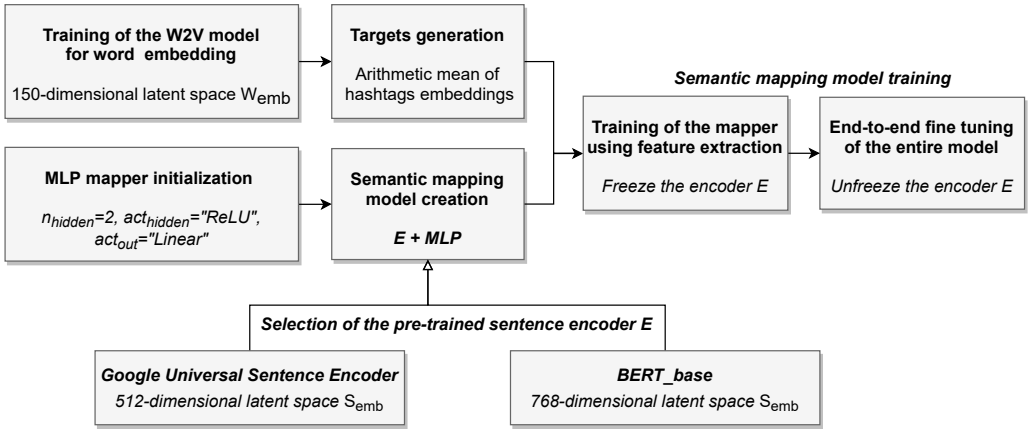


Fig. 2: Training of the *W2V* model for word embedding and target generation. Creation of the semantic mapping model (encoder (*E*) + mapper (*MLP*)) and two-step training: training of the mapper using feature extraction and fine-tuning of the entire model.

The input of this step is composed of the set of posts P and the selected encoder E exploited for computing a representation of a given post $p \in P$ within the space S_{emb}. We tested two different pre-trained models for sentence embedding, described in Section 2, namely GUSE and BERT. The output is the HASHET model $\mathcal{M}$.

ALGORITHM 1: Semantic mapping model creation and training.

Input : Set of posts P, selected encoder E
Output: HASHET model $\mathcal{M}$

1 $P \leftarrow preprocess\_data(P)$;
2 /* Word2Vec training and target vectors generation */
3 $W2V \leftarrow Word2Vec.train(P)$;
4 $targets \leftarrow \varnothing$;
5 **for** $p \in P$ **do**
6 $target(p) \leftarrow \varnothing$;
7 **for** $h \in H(p)$ **do**
8 $W_{emb}(h) \leftarrow W2V.embed(h)$;
9 $target(p) \leftarrow target(p) + W_{emb}(h)$;
10 $targets \leftarrow targets \cup \frac{target(p)}{|H(p)|}$;
11 /* Semantic mapping model creation */
12 $E \leftarrow init\_from\_pretrained()$;
13 $MLP \leftarrow init\_from\_scratch(n_{hidden} = 2,\ act_{hidden} = \text{“}ReLU\text{”},\ act_{out} = \text{“}Linear\text{”})$;
14 $SM \leftarrow stack(E, MLP)$;
15 /* Training of the MLP mapper using feature extraction */
16 $SM.E.freeze()$;
17 $opt \leftarrow ADAM(learning\_rate = 1e^{-3})$;
18 $SM.train(x = P,\ y = targets,\ loss = \text{“}cosine\_distance\text{”},\ optimizer = opt)$;
19 /* End-to-end fine tuning of the entire model */
20 $SM.E.unfreeze()$;
21 $opt \leftarrow ADAM(learning\_rate = 3e^{-5})$;
22 $SM.train(x = P,\ y = targets,\ loss = \text{“}cosine\_distance\text{”},\ optimizer = opt)$;
23 $\mathcal{M} \coloneqq \langle W2V, SM \rangle$;
24 **return** $\mathcal{M}$

The first step of the process (line 1) is to clean up data in order to prepare the corpus P for the embedding process. In particular, the input posts are modified and filtered by using a function $preprocess\_data(P)$, which performs the following operations:

— Posts with no hashtags are removed.
— Posts are cleaned using regular expressions for standardizing the text encoding into UTF-8, solving the problems related to the presence of characters of different encodings, and filtering out URLs.
— The text of each post is normalized by transforming it to lowercase and replacing accented characters with regular ones.
— Words are lemmatized and stemmed for allowing matches with declined forms (e.g., vote or votes or voted → vot).
— Stopwords are removed from text by using preset lists.
— Bigrams are found in the corpus for better capturing semantics using the Phrases module of the Gensim library (San Francisco → San_Francisco)

Afterwards, a Word2Vec model is trained on the pre-processed corpus P following the Continuous Bag-of-Words (CBOW) approach (line 3). Given a certain word w of the corpus, the $W2V$ model outputs a 150-dimensional vector $W_{emb}(w)$, which is a latent representation of the input word in the latent space W_{emb}. In this way, Word2Vec is exploited to capture the semantic relationships between hashtags and words, and hashtags themselves. As semantically similar hashtags are used in similar contexts, lying close together in the latent space, this induced clustering structure is exploited by HASHET for increasing its recommendation abilities. After the training of the

Word2Vec model, the target vectors for the semantic mapping phase are generated with respect to the embedding learned by the $W2V$ model in the W_{emb} space (lines 4-10). Specifically, an empty list $targets$ is initialized (line 4) and filled with the target vector of each post p of P. Given the current post p and the set of its hashtags $H(p)$, the latent representation of each hashtag h_p in $H(p)$, $W_{emb}(h_p)$ is computed (line 8). Then, the target vector for p, $target(p)$, is obtained as the arithmetic mean of all $W_{emb}(h_p)$ and added to the list (lines 9-10). It can be seen as a summarizing concept about the hashtags in $H(p)$, and can be written as follows:

$$target(p) = \frac{1}{|H_p|} \sum_{h_p \in H_p} W_{emb}(h_p) \tag{1}$$

HASHET is based on the translation between sentences and words/hashtags domain, i.e. the mapping between the latent representation of the entire post in the semantic space S_{emb} and its hashtags condensed in the corresponding target vector embedded in W_{emb}. Therefore, a crucial point of the model is the projection of the embedded sentences lying in S_{emb} into the words/hashtags latent space W_{emb}. Specifically, this mapping of the semantic vectors is learned using a semantic mapping model SM, composed by stacking two main blocks: the pre-trained sentence encoder E and the mapper MLP (line 14). The encoder E is initialized by loading its pre-trained weights (line 12), while the MLP mapper is created from scratch (line 13), by initializing a multi-layer perceptron with two hidden layers. In particular, $\mathcal{H}^{(1)} = 350$ and $\mathcal{H}^{(2)} = 250$ are the number of neurons in the first and the second hidden layer, while the output layer has 150 neurons, as it determines a 150-dimensional vector lying in W_{emb}. For what concerns the activation functions we used the Rectified Linear Unit ($ReLU$), defined as $ReLU(x) = x^+ = max(0, x)$, in the two hidden layers and a linear activation (lin), defined as the identity function, for the output layer. The $ReLU$ activation was used to introduce non-linearity in the mapping process; this choice was driven also by its interesting properties, such as sparse activation, scale invariance and efficiency.

Given a post $p \in P$, its semantic representation $S_{emb}(p)$ is computed by the first block of the SM model, i.e. the encoder E, obtaining a σ-dimensional semantic representation vector, where σ is the dimensionality of the sentence embedding space S_{emb}. Then, this latent representation of the input post p is fed to the MLP mapper, which outputs a 150-dimensional embedding vector lying in W_{emb}. The mapping process is driven by a cosine distance loss aiming at minimizing the angle between the projection into W_{emb} of the semantic vector $S_{emb}(p)$ and the condensed representation of its hashtags, $target(p) \in W_{emb}$. The loss $\mathcal{L}$ can be derived as follows:

$$h_j^{(1)} = ReLU\left(\sum_{i=1}^{\sigma} w_{ij}^{(1)} s_i + b_j^{(1)}\right),\ j = 1, ..., \mathcal{H}^{(1)},\ Semb(p) = s_1 ... s_\sigma \tag{2}$$

$$h_j^{(2)} = ReLU\left(\sum_{i=1}^{\mathcal{H}^{(1)}} w_{ij}^{(2)} h_i^{(1)} + b_j^{(2)}\right),\ j = 1, ..., \mathcal{H}^{(2)} \tag{3}$$

$$out_j = lin\left(\sum_{i=1}^{\mathcal{H}^{(2)}} w_{ij}^{(out)} h_i^{(2)} + b_j^{(out)}\right),\ j = 1, ..., 150 \tag{4}$$

$$\mathcal{L}(S_{emb}(p), target(p)) = cosine\_distance(target(p), out) \tag{5}$$

where:

— $h^{(1)}$ and $h^{(2)}$ are the outputs of the first and the second hidden layer, while out is the result of the output layer, which determines the 150-dimensional predicted vector.
— $W^{(1)} \in \mathcal{R}^{|S_{emb}| \times \mathcal{H}^{(1)}}$, $W^{(2)} \in \mathcal{R}^{\mathcal{H}^{(1)} \times \mathcal{H}^{(2)}}$, $W^{(out)} \in \mathcal{R}^{\mathcal{H}^{(2)} \times 150}$ are the weights to be learned in the first and the second FC-layer, and the output linear layer respectively.

The training of the SM model, implemented in Python using the high-level framework *Keras*[4] with *TensorFlow*[5] back-end, is divided in two steps:

(1) *Training of the mapper using feature extraction*. In this step we used transfer learning for training the SM model, by freezing the encoder E (line 16), which means that its weights will not be changed during training. This way, only the mapper MLP, composed of the top layers of SM, will be trained with the pairs $\langle S_{emb}(p), target(p) \rangle \ \forall p \in P$ (line 18), while the encoder E is used as a feature extractor for computing $S_{emb}(p)$ for a given p. The used optimizer is ADAM [Kingma and Ba 2014], initialized with the default learning rate $1e^{-3}$ (line 17).
(2) *End-to-end fine tuning of the entire model*. After the mapper was trained to convergence, we incrementally adapted the pre-trained features of the encoder E to our translation task. This was achieved by fine-tuning the entire SM model on the pairs $\langle p, target(p) \rangle \ \forall p \in P$ (line 22), after having unfreezed the encoder E (line 20). In this step we used a very low learning rate of 3^{e-5} (line 21), as we only want to readapt the pre-trained features to work with our task and therefore large weight updates are not desirable at this stage, which also lowers the risk of overfitting.

At the end of the described process, the algorithm returns the HASHET model $\mathcal{M}$, defined as the pair $\langle W2V, SM \rangle$ (lines 23-24), used for the recommendation step 4.2.

4.2. Hashtags recommendation by latent space inspection and semantic expansion

In this step, the HASHET model, defined as the pair $\langle W2V, SM \rangle$, is used for recommending a consistent set of hashtags for a given post p. The different steps involved in this process are shown in Figure 3 and described by Algorithm 2.

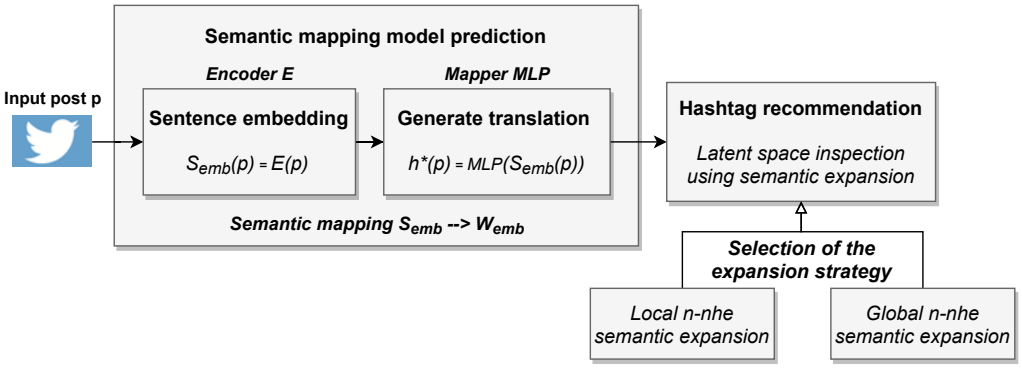


Fig. 3: Hashtag recommendation for a given post p, composed of two steps: 1) Semantic mapping of p exploiting the SM model (encoder + mapper) for obtaining the target vector $h^*(p)$; 2) latent space inspection using a selected semantic expansion strategy.

[4] https://keras.io/
[5] https://www.tensorflow.org/

The input is composed of: the post p, the HASHET model $\mathcal{M}$, the number of hashtags to recommend k, the expansion factor n, and the expansion strategy nhe. The output is the set of recommended hashtags $T^{k,n}(p)$ for input post p.

ALGORITHM 2: Hashtag recommendation by latent space inspection.

Input : post p, HASHET model $\mathcal{M} \coloneqq \langle W2V, SM \rangle$, ranked output limit k, expansion factor n, expansion strategy nhe
Output: set of recommended hashtags $T^{k,n}(p)$ for input post p

1 $h^*(p) \leftarrow \mathcal{M}.SM.predict(p)$;
2 $T^{k,n}(p) \leftarrow \varnothing$;
3 **if** *nhe is local* **then**
4 $\quad N^k(h^*(p)) \leftarrow \mathcal{M}.W2V.nearest\_hashtags(h^*(p), k)$;
5 $\quad T^{k,n}(p) \leftarrow N^k(h^*(p))$;
6 $\quad$ **for** $h \in N^k(h^*(p))$ **do**
7 $\quad\quad N^n(h) \leftarrow \mathcal{M}.W2V.nearest\_hashtags(h, n)$;
8 $\quad\quad T^{k,n}(p) \leftarrow T^{k,n}(p) \cup N^n(h) \setminus T^{k,n}(p) \cap N^n(h)$;
9 **else if** *nhe is global* **then**
10 $\quad N^{k+n}(h^*(p)) \leftarrow \mathcal{M}.W2V.nearest\_hashtags(h^*(p), k+n)$;
11 $\quad T^{k,n}(p) \leftarrow N^{k+n}(p)$;
12 **return** $T^{k,n}(p)$

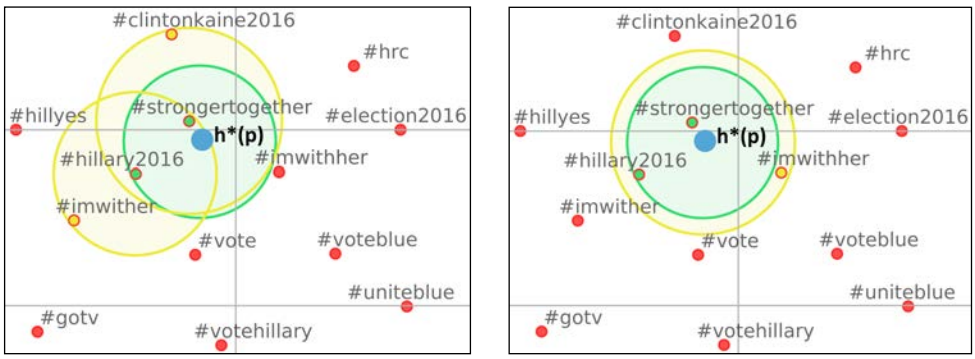
Given the input post p, the target vector vector $h^*(p)$ is obtained by using the semantic mapping model SM. In particular, when the *predict* function is called (line 1), the encoder block E of SM is exploited for computing the sentence embedding $S_{emb}(p)$ of the input post $p \in P$, which is then translated into the the corresponding 150-dimensional target vector $h^*(p) \in W_{emb}$ using the mapper block MLP. After the mapping, an empty set $T^{k,n}(p)$ is initialized (line 2), which will be filled with the recommended hashtags according to the selected expansion strategy (lines 3-11). In particular, if the $Local$ strategy is used (line 3), the set $N^k(h^*(p))$ is computed as the top-k nearest neighbors of $h^*(p)$ (line 4) and is assigned to $T^{k,n}(p)$ (line 5). Then, the set $T^{k,n}(p)$ is filled with the nearest hashtags of each hashtag in $N^k(h^*(p))$, removing duplicates (lines 7-8). If instead the $Global$ strategy is chosen (line 9), the set $N^{k+n}(h^*(p))$ is computed as the top-(k+n) nearest neighbors of $h^*(p)$ (line 10) and is assigned to $T^{k,n}(p)$ (lines 11). Finally, the algorithm returns the expanded set of vectors $T^{k,n}(p)$, which contains the hashtags recommended by the HASHET model.

Since there are many semantically related hashtags that are almost interchangeable, as they share the same meaning (i.e., #*trumptrain* and #*maga* or #*imwithher* and #*votehillary*), a semantic expansion based on the n-nearest hashtags (*n-nhe*) has been introduced, in order to capture semantic equivalences and maximize the match with the target hashtags. In particular, two different strategies have been proposed:

— *Local n-nearest hashtag expansion*. Given the expansion factor n, the set $N^k(h^*(p))$ is extended with the top-n nearest neighbors of each hashtag it contains.
— *Global n-nearest hashtag expansion*. Given the expansion factor n, the set $N^k(h^*(p))$ is extended by considering the top-(k+n) neighbors of $h^*(p)$, obtaining the extended set $N^{k+n}(h^*(p))$.

Therefore, the two strategies are aimed at expanding the set $N^k(h^*(p))$ composed of the k-nearest neighbors of the vector $h^*(p)$ ordered by likelihood, which is defined as the cosine similarity with respect to $h^*(p)$. The local approach can be considered a sort

of 2-hop decentralized process, where the distance is measured locally with respect to each hashtag in the non-expanded set. Thus we obtain the n-nearest hashtags for each neighbor of $h^*(p)$, where n is the expansion factor. Differently, when the global strategy is used, the nearest neighbor search process is extended by n steps maintaining the same center ($h^*(p)$). From this derives the adjective global, as every new hashtag vector is included according to its distance from $h^*(p)$, obtaining its $(k$+$n)$-nearest hashtags in the embedding space W_{emb}.



(a) local n-nhe (b) global n-nhe

Fig. 4: Local vs. global n-nhe expansion example (k=2 and n=1).

Figure 4 shows an example of how the two strategies work, with k=2 and n=1. The process starts from the 150-dimensional vector $h^*(p)$, obtained as the output of the mapping, represented by the blue point. Then, the two nearest neighbors of $h^*(p)$ are found obtaining the non-expanded set $N^k(h^*(p))$, containing the points highlighted in green, within the green circle. Starting from this set, the two strategies allow the inclusion of semantically related hashtags as described above, expanding by a factor n=1. These additive hashtags are represented by the points highlighted in yellow, within the yellow circles. The final set will be composed of the points located within the green and yellow circles. The resulting enriched set of vectors, referred to as $T^{k,n}(p)$, is the output of HASHET, and contains the suggested hashtags. By following this approach, the output set will be composed of semantically similar hashtags reflecting the semantic relationships learned in W_{emb} and the underlying topic-based clustering structure.

4.3. Why a translation approach? Exploit locality in the hashtag embedding space

The choice of modeling the problem as a translation task, is mainly related to how we modeled our target variable, a single 150-dimensional mean-pooled vector. This choice is based on the following assumptions:

(1) Tweets are short and a single tweet is very likely to talk about few topics (generally one). The same assumption can be found in [Zhao et al. 2011], where authors proposed Twitter-LDA, a topic model specifically designed for Twitter which treats tweets as single-topic.
(2) We assume that the hashtags embedding space is well-formed and the semantic relationships are well expressed inside it, which leads to the emergence of a topic-based clustering structure. In such a space, different hashtags which share semantic context will be highly clustered and will belong to the same topic.

Following the two assumptions above, instead of framing our problem as a multi-label classification, we used a translation-based approach, modeling our target variable using a fixed-length representation. This choice is good trade-off between efficiency and loss of information, since:

— Using a fixed-length representation (i.e., 150 in our case) is much more efficient with respect to the use of multi-label classification. In fact, as the number of possible hashtags gets larger, the size of the output layer increases together with the number of weights to be learned which leads to a higher cost of the training step.
— The loss of information caused by the use of this kind of representation is acceptable considering the above assumptions. In fact, according to (1), a given tweet is very likely to talk about a few related topics or even just one, so the hashtags it contains will share semantic context. Moreover, according to (2), the latent representations of these hashtags result highly clustered in the embedding space and taking their mean as a summarization will result in a vector lying in the same region.

At recommendation time, the system exploits a concept of locality in the hashtag embedding space that relies on: $i)$ the semantic relationships between the target vector and the candidate hashtags; $ii)$ the underlying topic-based clustering structure. Moreover, the recommendation abilities of the model can be improved by the semantic expansion process, further mitigating the negative effects of summarization.

5. PERFORMANCE EVALUATION

In this section we present the experiments carried out using the HASHET model on two different case studies. The first one concerns the 2016 US presidential election, characterized by the rivalry between Hillary Clinton and Donald Trump, while the second is related to the COVID-19 pandemic. In particular, for each case study we present the following analysis:

— An in-depth analysis of the word embedding space for highlighting the topic-based clustering structure induced by the hashtag distribution.
— An evaluation of performance varying the pre-trained encoder model (GUSE vs. BERT) and the semantic expansion strategy (local vs. global n-nhe).
— An extensive comparison with the most relevant state-of-art techniques.

Moreover, we investigated the ability of the HASHET model in discovering the main topic of a given tweet, starting from the set of recommended hashtags.

For evaluating the performance of the proposed model we used three different rank-based metrics: $precision$ ($P@k, n$), $recall$ ($R@k, n$) and F_1-$score$ ($F@k, n$). Given a post p and the set of its hashtags $H(p)$ (i.e., the target hashtags), the model outputs the set of recommended hashtags $T^{k,n}(p)$, where k is set equal to $|H(p)|$ and n is the expansion factor. We define a function $rel(t_i, p)$ for the i^{th} recommended hashtag $t_i \in T^{k,n}(p)$ such that $rel(t_i, p) = 1$ if $t_i \in H(p)$ so it is relevant for that post, $rel(t_i, p) = 0$ otherwise. Using this definition, the metrics can be written as follows:

$$P@k, n(p) = \frac{1}{|T^{k,n}(p)|} \sum_{i=1}^{|T^{k,n}(p)|} rel(t_i, p) \tag{6}$$

it is the fraction of successfully recommended hashtags among those suggested by the model.

$$R@k, n(p) = \frac{1}{|H(p)|} \sum_{i=1}^{|T^{k,n}(p)|} rel(t_i, p) \tag{7}$$

it is the hit rate of the model, or rather the fraction of target hashtags that have been successfully recommended.

$$F@k, n(p) = \frac{2 \times P@k, n(p) \times R@k, n(p)}{P@k, n(p) + R@k, n(p)} \tag{8}$$

it is the harmonic mean of precision and recall weighted by a β factor (i.e., $F_\beta,\ \beta = 1$). Furthermore, in our experiments, all tweets are grouped into five subsets, in relation to $|H(p)|$ (we impose the cardinality k of the non-expanded set $N^k(h^*(p))$ equal to $|H(p)|$) and scores are shown in relation to the increment of n. When n is equal to zero, any expansion is performed on neighbors, so $T^{k,n}(p)$ is equal to $N^k(h^*(p))$.

5.1. The 2016 US presidential election

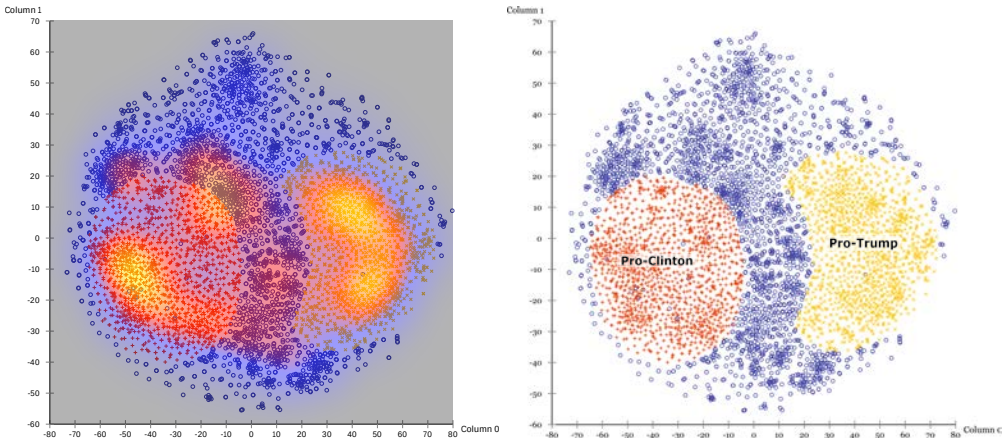
In this section we present the analysis carried out using HASHET on a corpus of about 2.5 millions tweets, posted by 521,291 users regarding the 2016 US elections, published from October 10, 2016 to November 7, 2016. The analysis has been performed on data collected for ten US swing states: Colorado, Florida, Iowa, Michigan, Ohio, New Hampshire, North Carolina, Pennsylvania, Virginia, and Wisconsin. Swing states are characterized by high political uncertainty, so they have been chosen in this analysis to capture a balanced corpus of posts with respect to the main topics of discussion, related to the support for the two candidates Hillary Clinton and Donald Trump.

The words/hashtags embedding space W_{emb} was obtained by training the CBOW Word2Vec model on the overall corpus, while the the semantic mapping model SM has been trained on a subset of 13,050 tweets published in New Hampshire: 9,787 of them have been used for learning the semantic mapping, while the remaining 3,263 make up the test set. We grouped test tweets in five classes according to the number of hashtags (from 1 up to 5) removing those containing more than 5 hashtags (115 tweets) for reducing noise. The obtained test set was composed of: 1637, 780, 439, 202 and 90 tweets, with 1,2,3,4 and 5 hashtags respectively (3,148 in total). The average number of hashtags per tweet is equal to 2 ($weighted\ avg. = 1.83$), in line with Twitter guidelines which recommend using no more than 2 hashtags per tweet as best practice[6].

5.1.1. Word embedding space analysis. A peculiar characteristic of microblogging posts in Twitter is that of associating the most common hashtags to a topic. So the hashtag projection of the semantic space of the word embeddings is expected to be highly clustered around the main discussion topics. In the following we show a series of representations of the latent space W_{emb} obtained by training the Word2Vec CBOW model during the analysis of this case study.

Since the 150-dimensional latent space W_{emb} can not be directly plotted, we firstly performed a dimensionality reduction using t-distributed stochastic neighbor embedding [Maaten and Hinton 2008], initialized through principal component analysis (PCA + t-SNE), to obtain a 2D representation of W_{emb}. Then, in order to identify dense groups of hashtags, we retained only the hashtags among the totality of latent representations, filtering out those with a frequency lower than 20. The resulting 2-dimensional latent space counts almost 5,000 hashtag points. Then, the OPTICS cut clustering algorithm [Ankerst et al. 1999] has been used to identify density-based clustering structures in this space and the results are shown in Figure 5. The cut-clustering algorithm was able to detect, consistently with the density estimation overlay (Figure 5(a)), two macro-clusters related to hashtags used for supporting the two major candidates Hillary Clinton and Donald Trump (Figure 5(b)). These clusters can be seen as the two macro-topics underlying the entire corpus. This topic-based separa-

[6] https://help.twitter.com/en/using-twitter/how-to-use-hashtags



(a) Density Estimation Overlay (b) OPTICS Cut Clustering

Fig. 5: OPTICS density-based cut-clustering structure of most frequent hashtags in the 2-dimensional representation of W_{emb} obtained through PCA + t-SNE.

tion of hashtags induced by the projection of their latent semantic distribution, can be seen better in Figure 6. The proposed scatter plot shows the 2-dimensional latent representation of the top three most frequent hashtags for the two candidates and their nearest neighbors. The considered hashtags are:

— Trump (yellow): *#maga, #trumptrain, #draintheswamp*
— Clinton (red): *#imwithher, #nevertrump, #strongertogether*

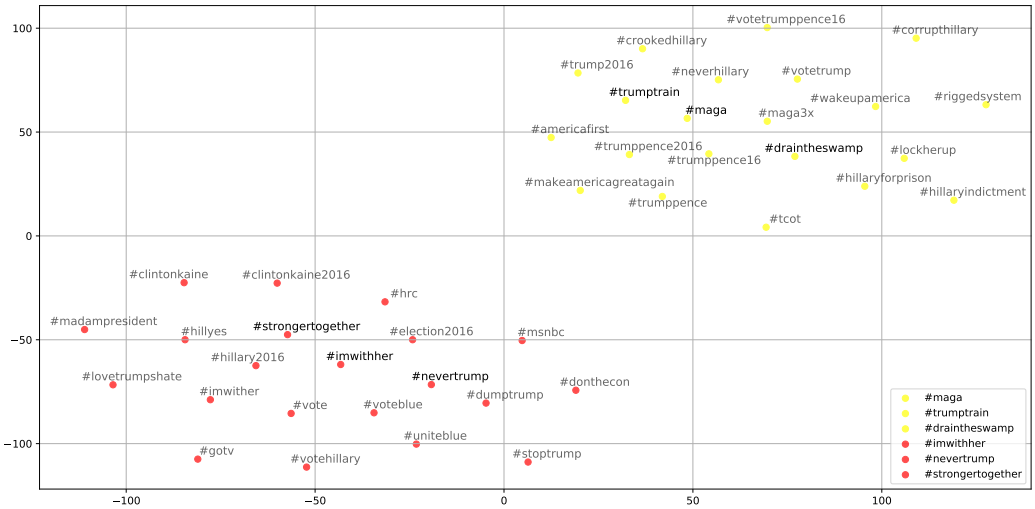
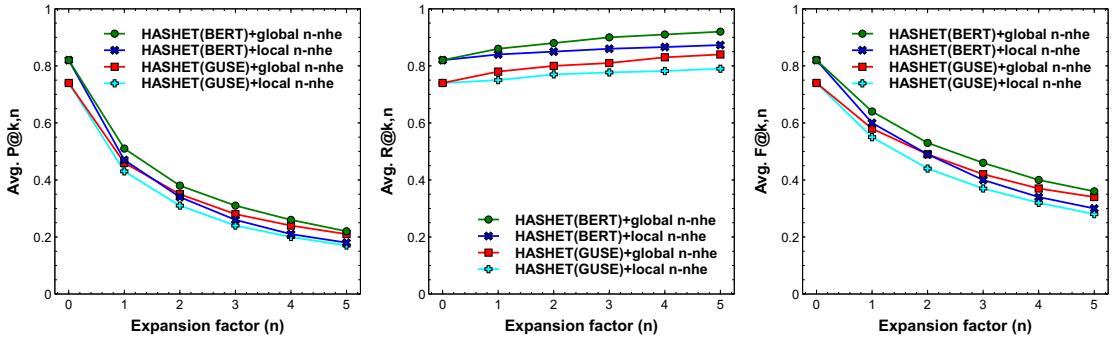


Fig. 6: Top 3 most frequent hashtags per candidate with their nearest neighbors.

The plot shows clearly the separation of hashtags according to the political polarization, that reflects the underlying topic structure identified by the clustering algorithm.

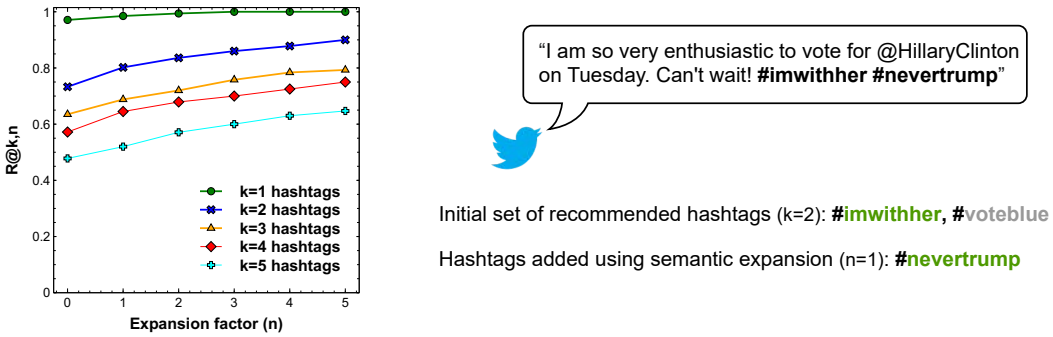
5.1.2. Encoding models and expansion strategies comparison. In this section we evaluated the use of different encoding models (*GUSE* vs. *BERT*) and semantic expansion strategies (*local* vs. *global*), showing the effects on the rank-based metrics. Figure 7 shows the benefits coming from the combined use of the BERT encoder and the global strategy, in terms of weighted precision, recall and F-score. Weighted averages are determined with respect to the scores achieved with different values of k (ranging from 1 to 5) and shown in relation to the increment of n (ranging from 0 (no expansion) to 5). We can observe that, for both semantic expansion strategies, the use of BERT leads to better recommendation results, which means that it can better grasp, compared to GUSE, the semantic aspects of a given tweet, producing more representative embeddings. On the other hand, for both encoders, the global expansion performs better than the local approach. This behavior is due to the higher importance given to the translation $h^*(p)$, which allows the global strategy to better exploit semantic relationships in the words/hashtags embedding space, by inspecting it with respect to a fixed center ($h^*(p)$). Differently, the local approach can lose this kind of information, focusing on the neighborhood of the top-k hashtags belonging to the non-expanded set $N^k(h^*(p))$. On the basis of these results we selected the BERT encoder and the semantic expansion strategy as the best configuration to be used in the following experiments.



(a) Weighted precision (b) Weighted recall (c) Weighted F-score

Fig. 7: Comparison of the two encoders (GUSE vs. BERT) and the two expansion strategies (global vs. local), in terms of precision, recall and F-score, weighted on k (number of target hashtags), varying n (expansion factor).

Afterwards, we analyzed the effects of global semantic expansion on the performance of HASHET in terms of $R@k, n$, which measures the recommendation hit rate of the model (Figure 8(a)). Firstly, we observed that the recommendation hit rate depends on the number of target hashtags (k), and decreases for increasing values of k. This is a common behavior among rank-based recommendation systems, where the difficulty in recommending (or retrieving) a group of items increases as the cardinality of the target set gets larger. Moreover, the plot shows how the expansion mechanism allows the model to recommend a more rich set of hashtags, by including additional ones that share semantic context with those contained in the non-expanded set. This aspect can be seen better in Figure 8(b), where we show an example of recommendation for a given tweet with two target hashtags: *#imwithher* and *#nevertrump*. The first target hashtag (*#imwithher*) is found among the top-k hashtag initially recommended, while the second (*#nevertrump*) is obtained through semantic expansion with $n = 1$.



(a) Hit rate for different values of k

(b) Recommendation example (k=2, n=1)

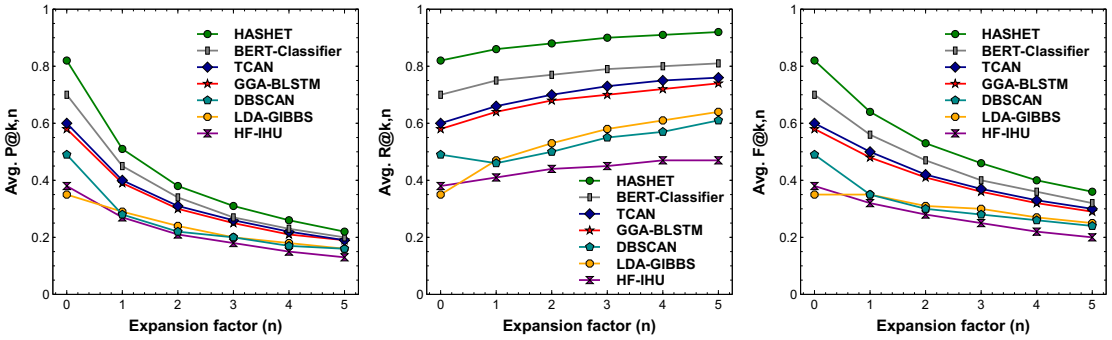
Fig. 8: Effects of semantic expansion on hit rate for different values of k, jointly using BERT and global n-nhe, and a recommendation example with k=2 and n=1.

5.1.3. Comparison to other methods. In order to evaluate the accuracy of HASHET, in both recommending a consistent set of hashtags and detecting the correct hashtag-based polarization of a given post, we carried out an extensive comparison with the most relevant techniques used in literature:

— Generative models:
  — LDA-GIBBS [Godin et al. 2013]. This method exploits the Latent Dirichlet Allocation and Gibbs sampling for finding out the underlying topic distribution, used for recommending general hashtags.
— Unsupervised models:
  — DBSCAN [Ben-Lhachemi and Nfaoui 2018]. This method is based on the embedded representation of Twitter microblog posts and performs the following steps: i) a given post is represented as the weighted average of its word embeddings; ii) latent representations of posts are clustered according to their syntactic and semantic similarity using a density-based approach; iii) top-k hashtags are found by computing the similarity between the entered post and the centroids of the obtained clusters.
  — HF-IHU [Otsuka et al. 2016]. The authors proposed a hashtag recommendation system for Twitter data streams based on a novel ranking scheme, the Hashtag Frequency-Inverse Hashtag Ubiquity (HFIHU). It consists of a variation of TF-IDF that considers hashtag relevancy and microblog data sparseness.
— Supervised models:
  — TCAN [Li et al. 2019]. This method exploits an attention based neural network to learn the representation of a microblog post. Specifically, the authors proposed a novel Topical Co-Attention Network (TCAN) that models content attention and topic attention simultaneously.
  — GGA-BLSTM. It consists in a degenerate version of the aforementioned TCAN model, which takes into account only the content in the attention mechanism. It can be seen as a standard Bi-directional LSTM model enhanced with global general attention [Luong et al. 2015].
  — BERT-Classifier. It consists of a fully fine-tuned BERT classifier obtained by stacking a softmax layer on top of the BERT-base transformer-encoder. Since we are coping with an extreme sparse input, whereby only few hashtags than

those possible (on average two) are actually present in a single tweet, we have configured the model as follows. It was trained using the cross-entropy loss and each target vector has been normalized by scaling it with a factor $1/h$, where h is the number of hashtags in the related post. This solution, already used in other works ([Mahajan et al. 2018; Joulin et al. 2016]), led to better performances for such a sparse input. We experimentally evaluated this aspect by testing BERT with a classical per-hashtag sigmoid output and a binary logistic loss, obtaining a significant performance degradation.

Figure 9 shows the results obtained by HASHET in comparison with the other related techniques for the hashtag recommendation task, in terms of weighted precision ($P@k, n$), recall ($R@k, n$) and F-score ($F@k, n$). Weighted averages are determined with respect to the scores achieved by each technique with different values of k (ranging from 1 to 5) and shown in relation to the increment of n, ranging from 0 (no expansion) to 5. As explained in Section 5.1.2, we imposed the cardinality k of the non-expanded set $N^k(h^*(p))$ equal to $|H(p)|$, that is the number of target hashtags, while n is the expansion factor. When n is equal to zero, any expansion is performed on neighbors, so $T^{k,n}(p)$ is equal to $N^k(h^*(p))$. We used the global n-nhe strategy for semantic expansion in HASHET, adapting this expansion strategy to the other techniques. In general, given k equal to $|H(p)|$ and $n \geqslant 0$, every model outputs the top-(k+n) hashtags.



(a) Weighted precision (b) Weighted recall (c) Weighted F-score

Fig. 9: Comparison with the most relevant related works, in terms of precision, recall and F-score, weighted on k (number of target hashtags), varying n (expansion factor).

Compared to the aforementioned techniques, HASHET turned out to be the most accurate in recommending the target hashtags, outperforming the competitors in terms of precision, recall and F-score. During the evaluation we found out what follows.

By considering the comparison between the HF-IHU, DBSCAN and LDA-Gibbs based methods, we observed that HF-IHU performs worse than the others. This indicates that the clustering structure of tweet embeddings learned by the unsupervised approach as well as the topic structure identified by the generative model, capture more semantic information than the simple frequency-based scoring technique, leading to more representative suggested hashtags.

In comparing these more traditional techniques (HF-IHU, DBSCAN and LDA) to the attention-based models based on neural networks, we observed a significant improvement in recommendation accuracy. The main reason behind these higher performances is the ability to learn an accurate latent representation of the microblog, rich

in semantic information, also exploiting the attention mechanism. In particular, the comparison between GGA-BLSTM and TCAN shows that topic information is useful in learning this kind of representation. For this reason, the topical co-attention model achieved slightly better performance with respect to the GGA-BLSTM, by jointly modeling content attention and topic attention simultaneously. Furthermore, we noticed that the fine-tuned BERT classifier achieved even more accurate results, in line with the most recent empirical improvements due to transfer learning with language models in a broad set of NLP tasks [Devlin et al. 2018].

Moreover, it is clear to observe that our model outperformed both traditional and attention-based models. Similarly to neural models, we exploited a semantic representation of the microblog, generated in our case by a transformer-based deep sentence embedding model. The key difference is in how this representation is used to recommend hashtags. In neural models, a softmax layer is generally used to output the probability distributions of all candidate hashtags. Then, top-k hashtags ordered by decreasing probability are returned in output. Differently, in HASHET, starting from the latent representation of a post in S_{emb} the target vector $h^*(p)$ in the words/hashtags space W_{emb} is predicted using the neural-based semantic mapping. Then, the top-k nearest hashtags of $h^*(p)$ are found and enriched using semantic expansion, obtaining an output set composed of semantically similar hashtags that could be possibly related to multiple topics. This kind of inspection process, centered in $h^*(p)$, exploits a concept of locality in W_{emb} that relies on the semantic relationships learned among hashtags and the underlying topic-based clustering structure.

It is also worth noting that HASHET is less dependent on tuning and parameters with respect to the majority of other techniques. The LDA model that exploits Gibbs sampling and topical co-attention network are sensitive to the number of topics and the number of topical words for each topic. These two parameters control, in the two models respectively, the topic discovery process and the topic-based information used in the attention mechanism. A wrong setting of these parameters could lead to the identification of a poorly representative topic structure or the introduction of noise in topical information. Another parameter-sensitive technique is the density-based clustering of the embedded representation of training tweets. This model uses the DBSCAN algorithm that is highly dependent from min_{pts} and ϵ parameters. A wrong estimate of min_{pts} or ϵ could lead to the identification of an unrepresentative clustering structure, which hinders the recommendation performances of the model.

5.2. Coronavirus (COVID-19)

After the presentation of the 2016 US presidential election, here we discuss the analysis carried out using HASHET on a corpus of 704,867 tweets regarding the COVID-19 pandemic [Lamsal 2020], published from December, 23 to December, 27 2020.

As in the first case study, the words/hashtags embedding space W_{emb} was obtained by training the CBOW Word2Vec model on the overall corpus, while the the semantic mapping model SM has been trained on a subset of 24,903 tweets: 18,678 of them have been used for learning the semantic mapping, while the remaining 6,225 tweets have been used as test set. Covid-related tweets have been grouped in five classes according to the number of hashtags (from 1 up to 5) removing those containing more than 5 hashtags (196 tweets) for reducing noise. The obtained test set was composed of: 3011, 1591, 764, 375 and 288 tweets, with 1,2,3,4 and 5 hashtags respectively (6,029 in total) and an average number of hashtags per tweet equal to 2 ($weighted\ avg. = 1,89$).

We analyzed the topic-based separation of hashtags induced by their latent semantic distribution, which leads to the emergence of a clustering structure in the W_{emb} space. In particular, we firstly projected the latent representations in a 2-dimensional space through dimensionality reduction jointly exploiting Principal Component Analysis and

t-distributed Stochastic Neighbor Embedding. Afterwards, by using the OPTICS algorithm, we identified a density-based clustering structure composed of 13 groups of hashtags, each related to a different topic, shown in Figure 10. In addition, the top-5 most frequent hashtags for each topic-based cluster are shown in Table II.

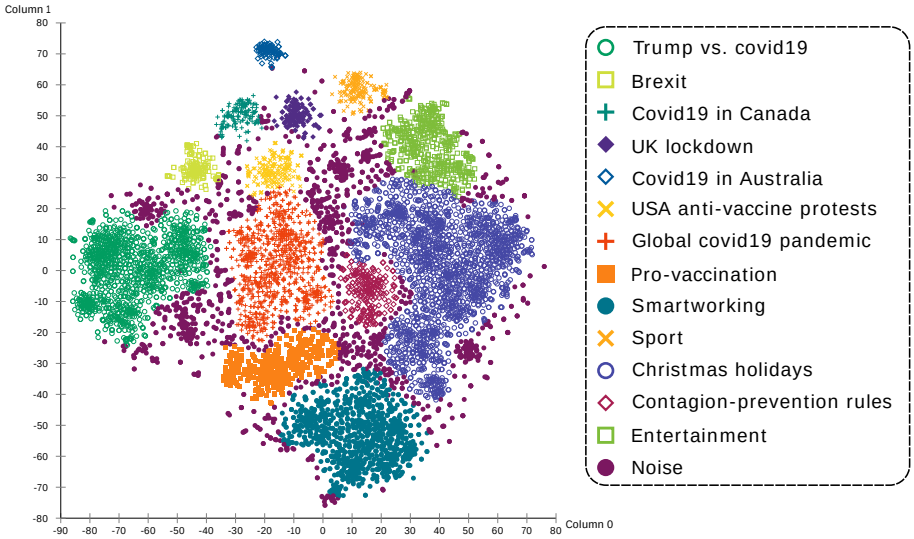


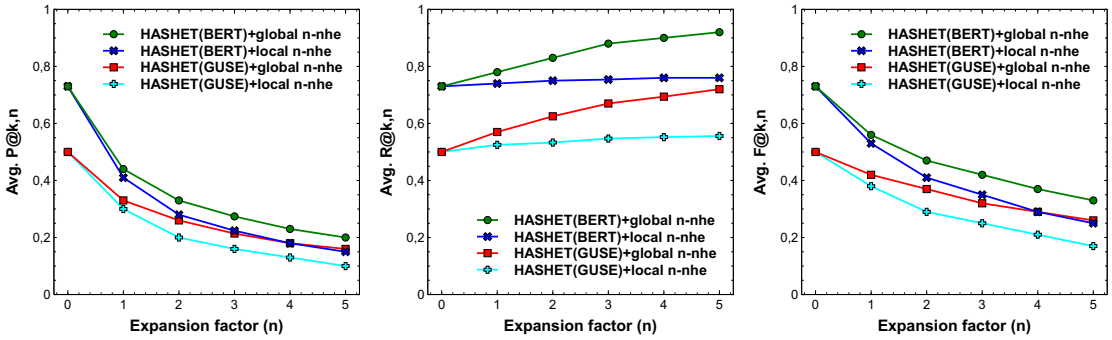
Fig. 10: OPTICS density-based cut-clustering structure in the 2-dimensional representation of W_{emb} obtained through PCA + t-SNE.

Topic	Top-5 most frequent hashtags
Global covid19 pandemic	#covid19, #coronavirus, #covid_19, #coronaviruspandemic, #travel
USA anti-vaccine protests	#losangeles, #california, #protests, #2019ncov, #florida
Christmas holidays	#christmas, #merrychristmas, #christmas2020, #covidchristmas, #christmaseve
Entertainment	#wonderwoman1984, #ww84, #starwars, #lightsforlouis,#happybirthdaylouistomlinson
Contagion-prevention rules	#wearamask, #stayhome, #staysafe, #socialdistancing, #washyourhands
Smartworking	#workfromhome, #jobs, #business, #wfh, #remotejobs
Pro-vaccination	#vaccine, #covidvaccine, #healthcare, #covid_19, #sarscov2, #frontlineheroes
UK lockdown	#covid19uk, #tier4, #coronavirusuk, #londonlockdown, #uklockdown
Covid19 in Canada	#canada, #cdnpoli, #onpoli, #bcpoli, #ontariolockdown
Sport	#browns, #nba, #nfl, #football, #rockets
Trump vs. covid19	#trump, #trumpvirus, #republicans, #foxnews, #dopeydon
Brexit	#brexit, #brexitdeal, #wearenotgoingaway, #borishasfailedthenation, #boristheliar
Covid19 in Australia	#7news, #sydney, #covid19aus, #covid19vic, #gladyscluster

Table II: Top-5 most frequent hashtags per topic.

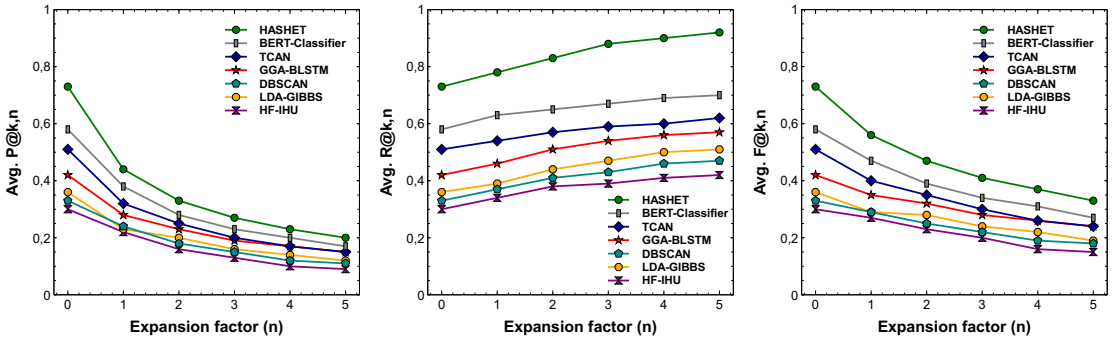
As for the first case study, we analyzed the performance of HASHET varying the pretrained encoder model (GUSE vs. BERT) and the semantic expansion strategy (local vs. global n-nhe). The results, shown in Figure 11, confirm the benefits coming from the combined use of BERT and global semantic expansion.

Even in this case study, HASHET has been compared with the most relevant related techniques described in Section 5.1.3, achieving the best recommendation results as shown in Figure 12. We observed that even if the results achieved by the different



(a) Weighted precision (b) Weighted recall (c) Weighted F-score

Fig. 11: Comparison of the two encoders (GUSE vs. BERT) and the two expansion strategies (global vs. local), in terms of precision, recall and F-score, weighted on k (number of target hashtags), varying n (expansion factor).



(a) Weighted precision (b) Weighted recall (c) Weighted F-score

Fig. 12: Comparison with the most relevant related works, in terms of precision, recall and F-score, weighted on k (number of target hashtags), varying n (expansion factor).

techniques are characterized by similar trends, this case study is more difficult then the first one, due to a larger set of discussion topics which leads to a more variegate set of hashtags. Moreover, the topic-based techniques, like LDA and TCAN, performed slightly better in this case study, albeit in proportion to its greater difficulty, thanks to the presence of richer topic information and their ability to effectively exploit it. Also the HASHET model benefits from this aspect, as it exploits locality in the hashtag embedding space, which presents a well-formed topic-based clustering structure.

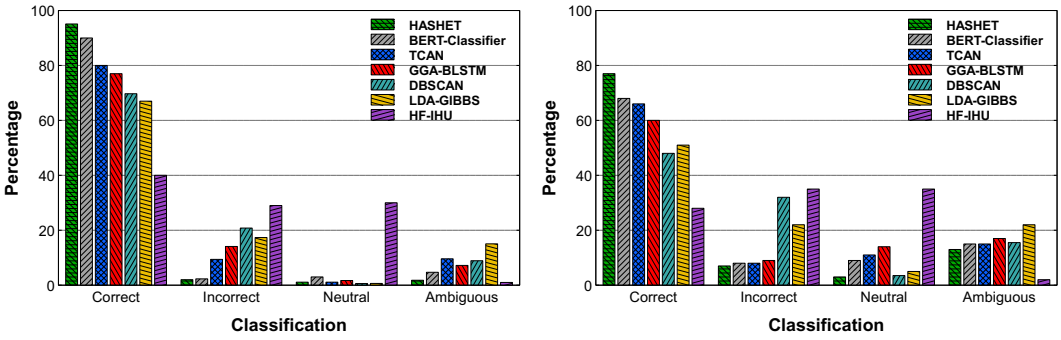
Compared to the other related techniques, HASHET turned out to be the most effective model for the hashtag recommendation task, outperforming either traditional techniques or neural-base models based on different types of attention mechanisms, such as topical co-attention, general global attention or self-attention. These promising results fully confirm those achieved in the first case study and the effectiveness of HASHET even in the presence of different topics of discussion such as covid19, vaccination or smart working.

5.3. Topic discovery using hashtag recommendation

The massive amount of opinion-rich multi-modal data in microblogging can be effectively exploited for discovering the public opinion in a community of users, analyzing their interactions and modeling their perception of facts, events and public decisions [Belcastro et al. 2019]. For example, Twitter posts have been analyzed by several opinion mining techniques for estimating, starting from their hashtags, the polarization of public opinion on political events characterized by the competition of factions or parties [Belcastro et al. 2020; Marozzo and Bessi 2018]. A hashtag recommendation model could be used to predict hashtags for posts that do not have any, in order to enrich the data used in this kind of techniques. In this section, we investigated how the recommendation abilities of HASHET can be exploited for a topic discovery task, aimed at identifying the supported faction or the main topic of discussion. For this purpose, test tweets have been preliminarily classified as follows:

— If it contains hashtags from only one cluster, it is classified with the related label.
— If it contains hashtags from two or more clusters, it is classified as *ambiguous*.
— If it does not contain hashtags from any cluster it is classified as *neutral*.

As we only focus in this step on single-topic tweets, containing hashtags belonging to at most one cluster in W_{emb}, all the input tweets classified with a valid cluster label are given to the hashtag recommendation model as input. Then, the output set of hashtags is classified as explained above and the discovered topic is compared with the real label. Afterwards, a global semantic expansion is applied on those tweets classified as neutrals. In particular, the set of recommended hashtags is expanded using different values of n, iterating from 1 up to 5, and the process stops when a cluster label is assigned to the tweet, or if it is still neutral after the last iteration. Given a test tweet p, the actual topic determined from the set of its hashtags $H(p)$, is expected to be equal to the classification computed with respect to the recommended hashtags for that tweet, $T^{k,n}(p)$; otherwise it may be neutral, ambiguous, or incorrect. This last is the most worrying error in the case of political polarization, as the post is considered in favor of the opposite candidate. Figure 13 shows the results obtained by HASHET in comparison with the other related techniques for the detection of the political polarization of a given tweet, Clinton vs. Trump, and the main topic of discussion among those previously described in Table II related to COVID-19 pandemic.



(a) USA 2016: polarization discovery

(b) COVID19: topic discovery

Fig. 13: Comparison with the most relevant related work in detecting the hashtag-based topic of discussion.

Test tweets have been classified as *correct*, *incorrect*, *neutral* and *ambiguous*, as explained above; moreover, we adapted the global expansion process to the other state-of-art techniques. To summarize, we found the following.

Comparing the HF-IHU, LDA-Gibbs and the DBSCAN-based models, the first achieved very poor results, with the lowest amount of correctly classified tweets and a large amount of incorrect and neutral tweets, while the LDA-Gibbs and the DBSCAN-based models achieved better results, similar to each other, showing their ability in detecting an underlying topic and clustering structure respectively. The attention-based neural models (GGA-BLSTM, TCAN, BERT-Classifier) achieved higher performances, thanks to the ability to learn a representative embedding of the analyzed microblogs, capturing a lot of semantic information. In particular, TCAN showed slightly better performances with respect to the GGA-BLSTM, exploiting the topical information within the co-attention mechanism. Moreover, the BERT-Classifier outperformed both TCAN and GGA-BLSTM thanks to a better understanding of the semantic content of a given tweet, which confirms the effectiveness of transfer learning from language representation models. HASHET outperformed the other recommendation models in discovering the main topics of discussion, achieving both the highest percentage of correct and the lowest amount of incorrect classifications. These results confirm the ability of the model in determining a highly representative set of hashtags.

6. ON THE APPLICABILITY OF HASHET FOR REAL-TIME HASHTAG RECOMMENDATION

Hashtags continually evolve over time, linking social media content to a specific topic, event, theme, or conversation. In this section, we analyzed how HASHET can be exploited for real-time hashtag recommendation. The proposed model can be adapted as follows. The newly generated posts from the social media platform are collected in real-time, monitoring the importance of every detected hashtag. A trending hashtags map $\mathcal{T} = \langle h$: *hashtag*, d: *date*$\rangle$ is exploited, where the date d represents the moment in which the hashtag h becomes popular. A hashtag is considered popular (i.e., it is a *trending hashtag*) when it is presented to the system with a frequency higher than a fixed threshold. This can allow the capture of current trending topics. When a new popular hashtag is added to the system, the following process is triggered:

(1) *Update of the hashtag embedding space*. To be recommended, a hashtag must be encoded with a 150-dimensional vector within the W_{emb} embedding space. For this purpose, when a hashtag is detected as interesting and added to the $\mathcal{T}$ map, the hashtag embedding space must be updated by retraining the Word2Vec model.
(2) *Fine-tuning of the semantic mapping model*. Similarly to zero-shot learning models, HASHET is able to predict hashtags that are not present in the training corpus of tweets, thanks to the concept of semantic affinity and expansion. However, to better grasp the semantic aspects of the new hashtags, the semantic mapping model can be updated through fine-tuning, by freezing the encoder model and training the Dense layers of the MLP mapper with a small learning rate.
(3) *Temporal re-weighting of hashtags rank*. At recommendation time, made in date d^*, the hashtag ranking r, induced by the similarity with the target vector, can be exponentially weighted, based on the distance between d^* and $\mathcal{T}[h]$, for each candidate hashtag h. The weighting process is controlled by a decay factor λ, so higher decay values will result in a stronger penalization of the ranking. In particular, given a candidate hashtag h its new rank score $r^*(h)$ will be computed as follows:

$$r^*(h) = r(h) \times e^{-\lambda \times (d^* - \mathcal{T}[h])}$$

In this way we can suggest hashtags in line with the most current topics, while respecting the semantic relationships in the embedding space.

Table III describes the computational complexity of the HASHET model for the real-time hashtag recommendation task, where s is the length of the input sequence, $\mathcal{H}^{(i)}$ is the number of neurons of the i-th MLP fully-connected layer, V_h is the number of distinct hashtags in W_{emb}, k is the cardinality of the non-expanded set, n is the expansion factor and $|T^{k,n}(p)|$ is the cardinality of the expanded set (number of recommended hashtags). Specifically, we analyzed each step involved in the recommendation process for a given post, which is structured as described in Section 4.2 and refined with an additional re-weighting step of the hashtag rank.

Step	Sub-step	Complexity
Semantic mapping (per-layer complexity)	sentence encoding	$\mathcal{O}(s^2)$
	generate translation	$\mathcal{O}(\mathcal{H}^{(i)} \times \mathcal{H}^{(i-1)})$
Latent space inspection and semantic expansion	local n-nhe	$\mathcal{O}((k + k \times n) \times V_h)$
	global n-nhe	$\mathcal{O}((k + n) \times V_h)$
Real-time refinement	temporal re-weighting	$\mathcal{O}(\|T^{k,n}(p)\|)$

Table III: Computational complexity of the steps involved in the real-time recommendation process of HASHET.

In the following, we provide a concise description of the computational complexities shown in Table III.

— *Sentence encoding*. As stated in [Cer et al. 2018], the transformer model complexity is quadratic in sentence length, as each token attends to the others in the self-attention mechanism.
— *Generate translation*. The generation of the MLP i-th layer output involves a dot product operation with its weights matrix. Since fully-connected layers are used, the complexity is proportional to $\mathcal{H}^{(i)} \times \mathcal{H}^{(i-1)}$.
— *Local n-nhe*. The local strategy starts from the k nearest hashtags of the target vector, finding the n nearest hashtags for each one of them. The search of each nearest neighbor involves the calculation of the cosine similarity for every candidate hashtag lying in W_{emb}, so it is linear in V_h.
— *Global n-nhe*. The global strategy directly constructs the set of $k+n$ nearest hashtags of the target vector, which involves $k + n$ linear searches in W_{emb}.
— *Temporal re-weighting*. This operation is performed on all recommended hashtags, so the computational complexity of this step is linear in the cardinality of the set of hashtags finally recommended.

7. CONCLUSIONS

This paper proposes a hashtag recommendation model, called HASHET, based on two different latent spaces, where sentences and words/hashtags are embedded. The crucial point of the model consists in the semantic mapping of the latent space of sentences into the embedding space of hashtags, performed by using a feed forward neural network. The top-k recommended hashtags are determined by latent space inspection, taking the k-nearest neighbors of the projection in the words/hashtags latent space of the embedded sentence, enriching the obtained set using semantic expansion. We evaluated the effectiveness of two language models for sentence embedding and tested different semantic expansion strategies, finding out that the combined use of BERT and global n-nhe leads to the best recommendation results. We also analyzed the applicability of HASHET to a real-time scenario and a multilingual context.

In order to assess the effectiveness of HASHET, it has been applied to two real-world case studies related to the 2016 United States presidential election and COVID-19 pandemic. By jointly exploiting BERT and global expansion, HASHET achieved an average F-score up to 0.82 and a hit-rate up to 0.92 for hashtag recommendation and an accuracy of 95% for topic discovery. Furthermore, it significantly outperformed different competitive state-of-art methods (generative models, unsupervised models and attention-based supervised models), with an up to 15% improvement in F-score for the hashtag recommendation task and 9% for the topic discovery task.

As future work, several embedding techniques and expansion strategies can be investigated, in order to adapt the model to other scenarios and evaluate its effectiveness in different application domains. Moreover, the analysis on the applicability of HASHET for real-time hashtag recommendation can be further extended for making the model able to cope with the continuous evolution of hashtags on microblogging platforms.

REFERENCES

Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. 1999. OPTICS: ordering points to identify the clustering structure. *ACM Sigmod record* 28, 2 (1999), 49–60.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

Loris Belcastro, Riccardo Cantini, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2019. Discovering Political Polarization on Social Media: A Case Study. In *The 15th International Conference on Semantics, Knowledge and Grids*. Guangzhou, China. To appear.

Loris Belcastro, Riccardo Cantini, Fabrizio Marozzo, Domenico Talia, and Paolo Trunfio. 2020. Learning Political Polarization on Social Media using Neural Networks. *IEEE Access* 8, 1 (2020), 47177–47187.

Nada Ben-Lhachemi and El Habib Nfaoui. 2018. Using tweets embeddings for hashtag recommendation in Twitter. *Procedia Computer Science* 127 (2018), 7–15.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Lyn Untalan Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder. In *In submission to: EMNLP demonstration*. Brussels, Belgium. https://arxiv.org/abs/1803.11175 In submission.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

Shi Feng, Yang Wang, Liran Liu, Daling Wang, and Ge Yu. 2019. Attention based hierarchical LSTM network for context-aware microblog sentiment classification. *World Wide Web* 22, 1 (2019), 59–81.

Fréderic Godin, Viktor Slavkovikj, Wesley De Neve, Benjamin Schrauwen, and Rik Van de Walle. 2013. Using topic models for twitter hashtag recommendation. In *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 593–596.

Yeyun Gong, Qi Zhang, and Xuanjing Huang. 2018. Hashtag recommendation for multimodal microblog posts. *Neurocomputing* 272 (2018), 170–177.

Matthew L. Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient Natural Language Response Suggestion for Smart Reply. *CoRR* abs/1705.00652 (2017). http://arxiv.org/abs/1705.00652

Jiajia Huang, Min Peng, and Hua Wang. 2015. Topic detection from large scale of microblog stream with high utility pattern clustering. In *Proceedings of the 8th Workshop on Ph. D. Workshop in Information and Knowledge Management*. ACM, 3–10.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 1681–1691.

Armand Joulin, Laurens Van Der Maaten, Allan Jabri, and Nicolas Vasilache. 2016. Learning visual features from large weakly supervised data. In *European Conference on Computer Vision*. Springer, 67–84.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Ryan Kiros, Yukun Zhu, Ruslan Salakhutdinov, Richard S. Zemel, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2015. Skip-Thought Vectors. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2 (NIPS'15)*. 3294–3302.

Abhay Kumar, Nishant Jain, Suraj Tripathi, and Chirag Singh. 2019. From Fully Supervised to Zero Shot Settings for Twitter Hashtag Recommendation. *arXiv preprint arXiv:1906.04914* (2019).

Rabindra Lamsal. 2020. Coronavirus (COVID-19) Tweets Dataset. (2020). DOI:http://dx.doi.org/10.21227/781w-ef42

Yang Li, Ting Liu, Jingwen Hu, and Jing Jiang. 2019. Topical Co-Attention Networks for hashtag recommendation on microblogs. *Neurocomputing* 331 (2019), 356–365.

Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. 2016. Hierarchical question-image co-attention for visual question answering. In *Advances in neural information processing systems*. 289–297.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. 2018. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 181–196.

Fabrizio Marozzo and Alessandro Bessi. 2018. Analyzing Polarization of Social Media Users and News Sites during Political Campaigns. *Social Network Analysis and Mining* 8, 1 (2018), 1–13. ISSN:1869-5469.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* (2013).

Eriko Otsuka, Scott A Wallace, and David Chiu. 2016. A hashtag recommendation system for twitter data streams. *Computational social networks* 3, 1 (2016), 3.

Junbiao Pang, Fei Jia, Chunjie Zhang, Weigang Zhang, Qingming Huang, and Baocai Yin. 2015. Unsupervised web topic detection using a ranked clustering-like pattern across similarity cascades. *IEEE Transactions on Multimedia* 17, 6 (2015), 843–853.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual bert? *arXiv preprint arXiv:1906.01502* (2019).

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. (2018).

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomáš Kočiskỳ, and Phil Blunsom. 2015. Reasoning about entailment with neural attention. *arXiv preprint arXiv:1509.06664* (2015).

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

Jieying She and Lei Chen. 2014. Tomoha: Topic model-based hashtag recommendation on twitter. In *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 371–372.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*. 5998–6008.

Wayne Xin Zhao, Jing Jiang, Jianshu Weng, Jing He, Ee-Peng Lim, Hongfei Yan, and Xiaoming Li. 2011. Comparing twitter and traditional media using topic models. In *European conference on information retrieval*. Springer, 338–349.