Time-aware Adaptive Tweets Ranking through Deep Learning

Carmen De Maio[a], Giuseppe Fenza[b], Mariacristina Gallo[b], Vincenzo Loia[b,*], Mimmo Parente[b]

[a] *Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica Applicata, University of Salerno, 84084 Fisciano (SA), Italy*
[b] *Dipartimento di Scienze Aziendali - Management & Innovation Systems, University of Salerno, 84084 Fisciano (SA), Italy*

Abstract

Generally, tweets about brands, news and so forth, are mostly delivered to the Twitter user in a reverse chronological order choosing among those twitted by the so-called followed users. Recently, Twitter is facing with information overload by introducing new filtering features, such as "while you are away", in order to show only a few tweets summarizing the posted ones, and ranking the tweets considering the quality, in addition to timeliness. Trivially enough we state that the strategy to rank the tweets to maximize the user engagement and, why not, augmenting the tweet and re-tweet rates, is not unique. There are several dimensions affecting the ranking, such as time, location, semantic, publisher authority, quality, and so on. We point out that the tweet ranking model should vary according to the user's context, interests and how those change along the timeline, cyclically, weekly or at specific date-time when the user logs in.

In this work, we introduce a deep learning method attempting to re-adapt the ranking of the tweets by preferring those that are more likely interesting for the user. User's interests are extracted by mainly considering previous user re-tweets, replies and also the time when they occurred.

We evaluate a ranking model by measuring how many tweets that will be

*Corresponding author. Tel.: +39-089-968228
URL: loia@unisa.it (Vincenzo Loia)

re-tweeted in the near future were included in the top-ranked tweet list. The results of the proposed ranking model revealed good performances overcoming the methods that consider only the reverse-chronological order or user's interest score. In addition, we pointed out that in our dataset the most impacting features on the performance of proposed ranking model are: publisher authority, tweet content measures, and time-awareness.

Keywords: Learning to Rank, Twitter, Deep Learning, Deep Neural Network.

---

1. Introduction

Context. Nowadays, we are assisting to a social data explosion. Facebook and Twitter are very popular communication platforms so that they are playing an important role in cultural, social, and political events. Social networking is a core part of the online experience [1]. Nevertheless, tons of tweets are daily posted, thousands of them happen every second and people are overwhelmed by the incoming information. Posts are authored by anyone from wherever around the world, and so, Twitter and Facebook have become attractive for spammers [2] compromising also the worth of the information source. The provisioning of the valuable tweet at the right time requires facing with information overload problem introducing filtering and ranking methods considering the user's interests, the activity that he/she is performing, the quality and relevance of the content, and so on.

In general, tweets are mostly delivered to the user in a reverse chronological order by considering ones that are published by the followed users. Recently, Twitter is facing with information overload proposing a new version of its timeline that ranks tweets by considering also the quality[1] and the relevance[2] in addition to the timeliness as stated in the official blog. New features are available on Twitter to show you relevant tweets list "in case you missed it", to give you a

[1] https://blog.twitter.com/2016/an-improved-timeline-for-consumers-and-brands

[2] https://blog.twitter.com/2016/moving-top-tweet-search-results-from-reverse-chronological-\order-to-relevance-order

subset of tweets based on their popularity, and how you interact with the tweet publisher. From the research point of view, some works are dealing with information overload on Twitter by defining tweets recommendation algorithms [3, 4], personalized ranking [5], filtering, and summarization [6, 7, 8, 9, 10, 11], customized according to several criteria.

In this paper, we emphasize that there is no unique and optimal criteria to rank the tweets maximizing users engagement and, why not, augmenting tweet and re-tweet rates creating more live commentary and conversations. There are several dimensions affecting the ranking, such as time, location, semantic, interestingness, publisher authority, and so on. Their impact on the ranking algorithm changes according to the user's context, the day of the week, the period of the year, and so forth. Indeed, the preferences change not only for different users but also for the same user according to the context in which user is when he/she comes to social media (i.e., Twitter). In fact, the same user may prefer to be updated by reading breaking news coming from social media when he/she is having a break, or when he/she is watching TV. Some users may prefer tweets related to the sporting event, but only in the hours following football matches. Unlikely, they may prefer to know that something important is happening in the nearby whenever it happens, even if they are searching something else.

Problem Definition. Formally, given a time-stamped finite tweet stream $TW = \langle tw_1, tw_2, \ldots, tw_n \rangle$, with some related information about publisher authority and user u, the task goal is to identify a function to rank the tweets in TW from those that are more relevant for u considering his/her own history (tweets, re-tweets, follows, etc.). The resulting ranking model should be *adaptive*, *personalized* and *time-aware* considering that the user's interests may change along the timeline and depend on the current context when the user logs in Twitter.

Proposed Solution. To achieve the aforementioned goal we define a *learning to rank* algorithm to sort a set of tweets (sketched in Fig. 1). Actually, learning to rank is a research area intensively investigated and many algorithms have been proposed, and consequently used in several fields including information retrieval tasks, focused search engines, and more recently, they are being adopted also for tweets ranking or recommendation [12]. In literature, we can distinguish the following main supervised approaches [13]: pointwise, pairwise or listwise. The main limitation of these algorithms is that supervised learning powers on the availability of user's feedbacks about the ranking of items, which are not easy to collect. In this sense, the most promising and natural approach is the pairwise that requires users' feedback only to determine what are the users' preferences with respect to pairs of items (i.e., tweets) instead of complete rank lists of them. We adopted a pairwise approach in which user's preferences are implicitly expressed by re-tweets and replies that we interpret as pairwise comparisons with respect to other tweets, for example, those shown in reverse chronological order, that have not been mentioned by the user. Among others, the pairwise algorithms, such as RankNet [14] and its deep version [15], has revealed good performances in ranking web pages to improve web search experience. In this work, we adopt an algorithm inspired to SortNet [16], a ranking algorithm based on deep neural network to rank tweets including several features to represent user, content, publisher, and so on. The aim is to learn a function to evaluate the choices between two tweets, i.e., tw_i and tw_j. Given a pair of tweets $tw_i, tw_j \in TW$, the aim is to learn a preference function $P : TW \times TW \ \rightarrow \{>, <\}$ which evaluates the user's interests with respect to the pair of tweets, i.e. $tw_i > tw_j$, if tw_i should be preferred to tw_j, and $tw_i < tw_j$, vice versa.

Contributions. Unlike other application domains, for instance, web search where learning to rank algorithms have already been widely applied, the strong dynamic nature of the microblogging stresses the importance of the model re-adaptation.This work introduces a deep learning method for tweet ranking capable to re-adapt itself along the timeline and considering different tweet and user's interests. Time-awareness is implemented by using datetime of the tweets during the ranking model training. More precisely, the main contributions of the proposed research are:

- Definition of a learning to rank algorithm for tweets; in particular, we use a pairwise algorithm assuming that each re-tweet and/or reply represents a user's feedback expressing preference for that topic, the publisher's authority, and so forth;

- Integration of datetime of the tweet, re-tweet, or reply during the training phase in order to provide different ranking results considering the moment when user logs into the Twitter; in fact, the occurrence of user's interest may recur cyclically in a given time slot (e.g., weekend, evening, etc.);

- Implementation of a continuous learning giving new sample items as input tuples for training the ranking model at each time the user expresses his/her preference replying or re-tweeting something;

- Adoption of tweet content *wikification* to semantically categorize the posts by linking tweet text to Wikipedia articles; this practice enables us to use corresponding Wikipedia entities to characterize the user's topics of interest.

Experimental Results. Starting from the collected tweet stream, we adopt our framework to perform a personalized tweet rank simulating different accessing time slots, and we evaluate its precision by applying Mean Average Precision (MAP) and Normalized Discount Cumulative Gain (NDCG) metrics. Performances have also been evaluated by omitting some significant features (i.e., tweet publisher's authority, social relation between tweet author and user, and time-awareness) in order to estimate their impact on the method performance. We evaluate the tweets ranking improvement counting how many top-ranked tweets will be re-tweeted/replied in the near future with respect to the *ignored ones*. The experimental results reveal promising performance and confirm the unsuitability of a simply reverse chronological order. In addition, we point out that time features play an important role because ranking preferences improves by including time features in the learning phase and considering the time slot when the users log in Twitter.

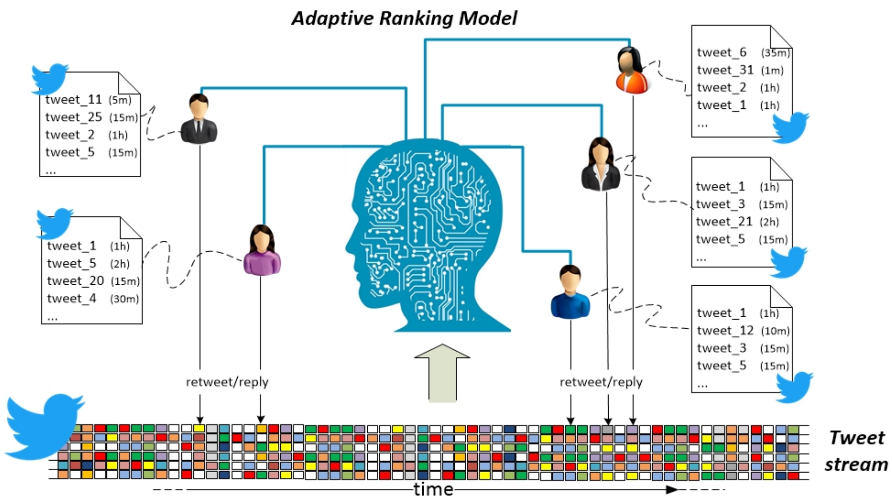

Figure 1: Overview of the proposed adaptive ranking model.

Outlines. The paper is structured as follows: Section 2 describes some related works; Section 3 discusses the deep neural network architecture used for the ranking model; Section 4 details the features selected to train the model and illustrates how the tweets components are modeled; then, in Section 5 the evaluation results are discussed; finally, the conclusion and future works close the paper.

2. Related Works

This section deals with the main relevant areas of related works: (1) ranking and recommendation in Twitter, (2) learning to rank with deep learning.

2.1. Ranking and Recommendation in Twitter

Recently, many research works are applying original recommendation and ranking methods to Twitter because the availability of such amount of information accessible in live streaming is a suitable test bench for experimenting novel methods. Most of these researches define recommendation algorithms on Twitter for suggesting tweets, as well hashtags, users to follow, and so forth. In

[17], a people-to-people recommender system is proposed taking into account the users' interests, sentiments, and attitudes, extracted from the tweets' contents. In [18], the authors present a novel followee ranking scheme using a latent factor model to leverage implicit users' feedback including both tweet content and social relation information for recommending high quality of top-k followees over microblogging systems. A modification of conventional social vector clocks has been previously proposed in literature to deal with friendship distance [19]. Location-related statuses are used for supporting information delivery in [20]. Hashtags recommendation technique is proposed in [21] by applying topic models and collaborative filtering techniques to assist users for retrieving content of interest.

In the regard of user's interest, most of the aforementioned works use topic models to project high-dimensional words into low-dimensional latent topics extracted from users' tweets and words are used to infer users' interests. Nevertheless, when dealing with short texts, like a tweet, there is a need to add neighbor documents for topic decomposition [22], [23]. In this work, we used tweet content wikification to automatically link named entities mentioned in the tweet to Wikipedia articles disambiguating the meaning. This practice was widely used [6], [24], [25] and seems to be not compromised by the short nature of the sentences.

Among others, tweets recommendation systems play a crucial role attempting to face with information overload in social media. Some works use social influence between friends for recommending tweet [11, 26]. The work presented in [27] proposes a model exploiting social phenomenon of homophily to achieve higher performance on both interest targeting and friendship prediction. In [3], the authors present a collaborative ranking model by considering as features tweet topic, social relation aspects, quality (i.e., using some content-based measures) of the tweet, publisher authority, etc., for recommending useful tweets to the users.

Some other works are adopting learning to rank methods to address tweets recommendation problem. SVMRank algorithm was adopted in [12] for generating a global ranking model to support information retrieval over microblogging using as input features relevance of the tweet content with respect to user's query, number of entities included in the tweet, and influence of authors of the tweet. Instead, a personalized ranking model using tweet history of the target user is defined in [28]. Our contribution adds to user and tweet related features the temporal ones for generating time aware ranking model, in fact, we argue that the engagement of the user with respect to the tweet content depends also on the time slot when the tweet pops out. We advise that time plays a crucial role to understand whether the user is cyclically (daily or weekly or at specific time) interested in a certain thing to carry out a more effective personalized and adaptive ranking model.

2.2. Learning to Rank with Deep Learning

Learning to rank algorithms found a lot of applicability ranging from web search to web services discovery [29], node ranking in the data center networks [30], Resource Description Framework properties ranking in the area of Semantic Web [31], and tweets recommendation. The applicability is wide so much that research works attempted to generalize ranking algorithm for training model in a source domain and apply it in another target domain in situation where no, or just some, labeled data are available [32, 33]. Unlikely, we have a lot of labeled data to use. In fact, the proposed work uses as labeled data the replies and re-tweets made by the user, that are essentially treated as explicit user feedbacks expressing pairwise preference among the arriving tweets.

A lot of research methods have been defined using Support Vector Machine [12], Neural Network [14], Random Forest [34], while we adopt a deep learning model. Recently, deep learning methods have been used in several application domains such as automatic speech recognition [35], image recognition [36], natural language processing [37]. Mirowski et al. [38] proposed text classification method customized for series of time-stamped documents (e.g., online news). In [39], a deep learning system for Twitter sentiment classification is proposed. Specifically, the research work proposed in [15] extends the ranking

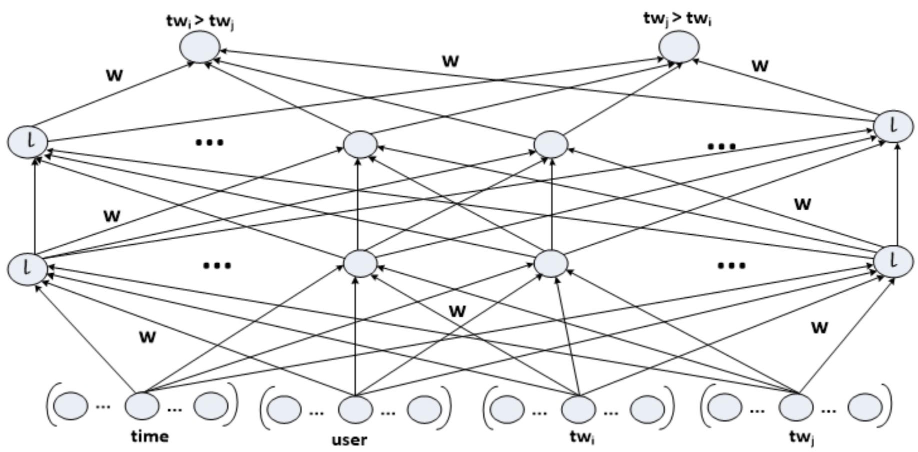

Figure 2: Deep Neural Network Architecture.

model of RankNet with deep architecture adapting the ranking model to each user analyzing search history and result preferences for supporting personalized search. Similarly, we propose a deep learning method to address adaptive and personalized learning to rank including features characterizing user's tweets, re-tweets, or reply.

Deep learning architecture that we propose is inspired to SortNet, introduced in [16], that is a comparative neural network architecture implementing pairwise learning to rank method that revealed good performance for the retrieval of documents in response to a query. Analogously to SortNet, we define a comparative multilayer perceptron feed-forward neural network to train comparison among tweets. The main distinguishing feature of the proposed model is that we experiment the inclusion of temporal features to carry out time-aware ranking results.

3. Deep Neural Network Architecture for Adaptive Tweet Ranking

The proposed method implements a pairwise preference learning where the function relies on a multilayer perceptron feed-forward neural network sketched in Fig. 2. Inspired to the Comparative Neural Network (CmpNN) introduced in [16], giving as input a couple of tweets (tw_i and tw_j), their temporal and user information, the neural network carries out the ranking relation between them i.e., $tw_i > tw_j$ (or $tw_i < tw_j$) as shown in Fig. 2. Temporal features allow us to train the system considering the time slot when the user's preference is expressed. Let us note that the user's preference for a tweet is essentially expressed by posting a re-tweet, or a reply to it. In particular, this tweet is preferred in contrast of those that arrived, but ignored, in the interval that goes from the time at which the tweet was originally posted, until the moment when the user's re-tweet or reply is posted. In addition, the user's features allow us to personalize the resulting ranking model. This aspect is related to the personalization of the resulting ranking model. Indeed, the users are represented by specifying topics of interest (see Section 4), thus the single user's feedback will impact on the ranking model for a class of similar users.

Specifically, the input layer is composed of 4 main components, i.e., the 4-tuple $\langle t, u, tw_i, tw_j \rangle$, in which:

- t represents the *date* and *time* in which the user has expressed an interest, i.e, a re-tweet or a reply;
- u represents the user;
- tw_i and tw_j represent the i-th and j-th tweet, respectively.

The feature sets of each component will be detailed in Section 4. The output layer includes 2 classes that determine which of the tweet tw_i, tw_j wins the comparison for the user u at the date and time t. In particular, given a user u and a couple of tweets tw_i and tw_j, according to u's interests at the time instance t, the label of the classes has value 0 for interest in tweet tw_i, and 1 for tw_j.

We consider some hidden layers. The number of layers and the configuration of hyper-parameters are based on the empirical observations [40]. In particular, the model is trained by setting the following hyper-parameters:

- Number of layers: 4 (i.e., 2 hidden layers);

- Learning rate: 0.1;
- Iterations number: 1200;
- The weight of each neuron is calculated by means of a Gaussian distribution. In particular, weights are, respectively: 0.042, 0.061, 0.1, 0.22.

The resulting comparative model is used into a classical sorting algorithm to rank the tweets for an arbitrary user when he/she logs in Twitter. A similar network configuration has also been trained when we exclude some features (as explained in Section 5.3) in order to understand their singular contribution. In particular, by changing the number of inputs we adjust number of iterations to 1000 (in this way we obtain a similar error score during the training) and the weight of the first neuron becomes 0.044. Such configurations have different impact on the training time (see Section 5.4).

4. Feature Selection for Adaptive Tweet Ranking

This section describes the set of features selected to represent each component of the 4-tuples $\langle t, u, tw_i, tw_j \rangle$ used in the defined deep neural network architecture.

The representation of the data-time, t (i.e., the re-tweet timestamp, see Section 3), consists of the day of the week and time slot. The data-time component and the granularity of time slot as well, are important to discover regularities in the data set about the moments when the user interact on Twitter. We opted to consider four time slots discriminating among morning, afternoon, evening and night.

The representation of the user, u, consists of a vector of frequencies of topics representing the level of interest with respect to a fixed set of categories derived by considering tweet content topics of the previous user's posts. The fixed set of categories is the same used to represent the tweet content topics detailed in Section 4.1.

Finally, the representation of the features of the tweets (i.e., tw_i or tw_j) consist of the following macro components:

- **Tweet Content Topics**: characteristic vector corresponding to the fixed set of categories covering the overall set of the tweet stream;
- **Quality and Popularity**: a set of features assumed to represent quality and popularity of the tweet;
- **Publisher's Authority**: features aiming to quantify authority of the publisher in terms of number of followers and the recency of his activity on Twitter.
- **Social Relations**: features that measure the social relationship between the user and the tweet publisher.

Let us note that the tweets and corresponding features have been extracted by means of Twitter Streaming API, whereas, social features about followers and friendship have been extracted by exploiting the findings about social graph shared from Kwak H. et al [41].

The following subsections detail the composition of the aforementioned features used to represent the tweets.

4.1. Tweet Content Topics

In this work we used *sentence wikification* [42] to semantically enrich tweet content representation by linking Wikipedia articles corresponding to the meaning of the sentence. Sentence wikification revealed to be not compromised by the short nature of the tweet [24, 6]. Given a tweet, tw_i, the sentence wikification service[3] retrieves $wiki(tw_i)$ that is a list of pairs $\left\langle topic_{i_k}, rd_{topic_{i_k}} \right\rangle$ where the first component is the Wikipedia article related to the tweet content and the second one is its specific relevance degree.

[3] Wikify service provided by the University of Waikato, publically available at `http://wikipedia-miner.cms.waikato.ac.nz/`. Let us note that we have exploited a local installation of the Wikipediaminer installation.

Since the granularity of Wikipedia articles characterizing the wikified tweets is too fine for our aim, we opt for using a fixed set of Wikipedia categories, $C = \{C_1, C_2, \ldots, C_m\}$, to which Wikipedia articles belong to. These categories enable us to obtain a more generic representation of the tweets. Then, for each tweet tw_i, we define its characteristic vector with respect to categories in C, where the membership is measured according to a given threshold of its relevance. More precisely each component of the array *tweet content topics* tct_{C_j} is determined as follows: $tct(tw_i) = \langle tct_{C_1}, tct_{C_2}, \ldots tct_{C_m} \rangle$.

$$tct_{C_j} = \begin{cases} 1 & \textit{iff} \; \exists \left\langle topic_{i_k}, rd_{topic_{i_k}} \right\rangle \in wiki(tw_i) | topic_{i_k} \in C_j \text{ and } rd_{topic_{i_k}} \geq \tau \\ 0 & otherwise \end{cases} \tag{1}$$

where τ is a fixed threshold that we set to 0.6. We empirically fixed the set C composed of 9 most representative Wikipedia categories for our tweet stream, that are:

C_0 = Weather, C_1 = Geography, C_2 = Sports, C_3 = Games, C_4 = Politics, C_5 = Film, C_6 = Music, C_7 = Currency, C_8 = Information Technologies (IT).

For example, let us consider the following tweet:

"Musicians making music *#piano*"

the sentence wikification service retrieves the following list of topics:

$$\langle Piano, 0.8 \rangle, \langle Music, 0.7 \rangle, \langle Musician, 0.7 \rangle$$

Then, since all of the extracted topics belongs to only one Wikipedia category among the fixed ones, i.e., "Music", the resulting vector will contain the 1 value at index position corresponding to that category:

$$tct = \{0, 0, 0, 0, 0, 0, 1, 0, 0\}.$$

4.2. Quality and Popularity

The quality and popularity of the tweet are beyond the specific interest shown by the user (e.g., a risk attack in a neighboring zone or a traffic restriction in a user's frequented area, and so on). In this work, these features are represented by the following measures:

- **Length**. Starting from the fact that the length of the tweet may impact the user's interest, this feature considers the total number of words.

- **Hash-Tags**. The presence of hashtags makes the tweet more or less informative and useful. So, we consider the total number of hashtags.

- **URLs**. Tweets are limited in terms of allowed characters (i.e., 140), and authors are used to adding one or more URLs that, for example, point to the source of their information. Such property is considered as an additional quality measure, so, this feature considers whether or not the tweet contains at least one URL.

- **Re-tweets**. The number of the re-tweets that is an index of the popularity and usefulness of the tweet.

- **Likes**. The number of the times someone has expressed a positive feeling about the tweet is also considered a quality indicator.

4.3. Publisher's Authority

These features intend to measure the *reputation* of the tweet publisher. Typically, users may prefer to read tweets published by a more or less authoritative publisher and, at the same time, the tweet quality is considered directly proportional to the author's authority (i.e., an authoritative author is likely to post an interesting and useful tweet). So, the publisher authority is measured with the following properties:

- **Followers**: the number of followers.

- **Status**: the total number of user's tweets.

4.4. User's Relation

This set of features models the relationship between the user and the tweet publisher. The intuition is that users should be more or less interested in reading tweets posted by their friends, people they choose to follow or sharing common interests. The evaluated measures are:

- **Followee-based Similarity**. This feature measures the size of the intersection set between the sets of their followees;
- **Friendship**. yes/no feature that indicates if the user and the tweet publisher are friends. On Twitter, the friendship relation is often induced when two users follow each other [43].

5. Evaluation

To evaluate the proposed ranking method, we collected a tweet stream and calculated the selected features described in Section 4 in order to prepare the training sample to build the ranking model. The test set is composed of the tweets that are adjacent to the stream used for training the model. Given a specific user, we tested the resulting ranking model evaluating the top-ranked tweets obtained by varying input time slot. The input time slot represents the moment when user logs in Twitter.

Several ranking measures are suitable to evaluate the results of learning to rank algorithms [44]. In particular, we adopted MAP [3] averaging on values of precision at n (P@n) [45], and NDCG [46] metrics.

Following subsections describe: the datasets used to train and test the ranking model (Section 5.1); the measures used to evaluate the performances (Section 5.2); and, finally, the obtained results (Section 5.3).

5.1. Dataset

In order to evaluate the proposed framework, we selected two random users and by means of the available social graph [41] we searched their followers and followees. From this subset of users, we extracted the first 5'000 users and captured their activity on Twitter. The original tweet stream is filtered to consider only the tweets posted in a period of two weeks (collected from 26/01/2017 to 12/02/2017) whose content is at least in one of the fixed set of categories (i.e., Weather, Geography, Sports, Games, Politics, Film, Music, Currency, Information Technologies (IT), see Section 4.1).

Table 1: Training-set statistics: tweets collected from 26/01/2017 to 10/02/2017

Training set					
	Morning	*Afternoon*	*Evening*	*Night*	***Total***
Re-tweets/Replies	1.595	1.619	2.055	1.453	**6.722**
Tweets	15.872	16.444	22.259	15.352	**62.987**
Users	314	316	340	275	**656**

Table 2: Test-set statistics: tweets collected from 10/02/2017 to 12/02/2017.

Test set					
	Morning	*Afternoon*	*Evening*	*Night*	***Total***
Re-tweets/Replies	17	15	28	69	**129**
Tweets	137	76	232	408	**835**
Users	13	12	18	29	**53**

Tables 1 and 2, detail the number of users, the number of tweets and the number of the corresponding re-tweets/replies grouped by time slot. Table 1 details the training set and Tables 2 refers to the test set used for evaluating the system. At the moment, we consider four different time slots regarding the re-tweet/reply time during the day: *Morning*, *Afternoon*, *Evening* and *Night*.

We collect a set of re-tweets/replies in the week-end subsequent to the training period and use them as positive samples. In particular, for each user, we selected the re-tweets/replies, and, for each one, we calculated the time interval between such tweet creation and the user re-tweet, and collect tweets published by the followees in this interval and neither re-tweeted nor replied, obtaining a list of N tweets arranged in a reverse chronological order.

Our dataset, as shown in Fig. 3, suffers of data sparsity: the number of users that frequently re-tweet/reply another tweet is very low. Since this aspect should negatively influence the resulting model, we considered some more contextual features during deep network training to generalize the training data as much as possible, as studied in [3]. For instance, we have included in the training

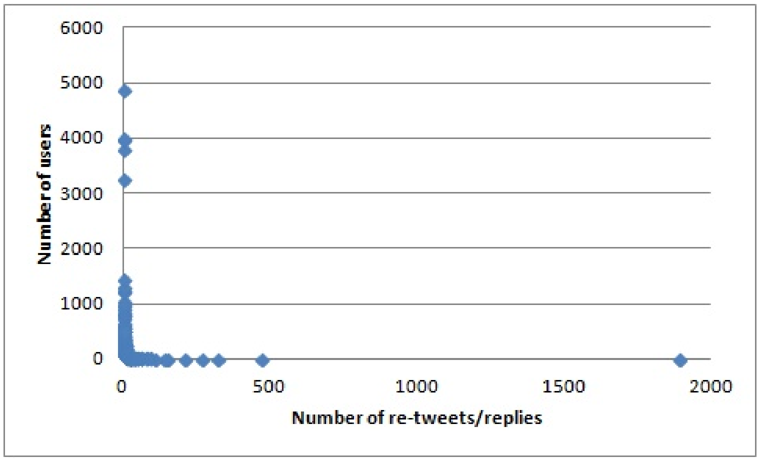

Figure 3: Sparsity of dataset in terms of number of re-tweet/reply(s) and number of users.

tuples user's information in order to represent not only individual (re)tweeting users but a class of them generalizing the resulting model. Analogously, we use tweet topic, number of re-tweets, and so on, for representing the tweets.

5.2. Metrics

Mean Average Precision and Normalized Discount Cumulative Gain are the measures used to evaluate the average of the ranking algorithm performance with respect to the overall users in the test set.

Given a user u, let us define $P@n$ that measures the relevance of the top n results of the ranking list:

$$P@n = \frac{relevant\ tweets\ in\ top\ n\ results}{n} \tag{2}$$

Given a user u, the average of the precision $P@n$ measured for all re-tweets/replies is Average Precision (AP_u) defined as follows:

$$AP_u = \frac{\sum_{n=1}^{N} P@n \cdot rel(n)}{N_u} \tag{3}$$

where N and N_u are, respectively, the number of tweets and of re-tweets/replies for the user u, $rel(n)$ is a function that has value 1 if the $n-$th tweet in the ordered list has been re-tweeted/replied by u, 0 otherwise. Thus, AP_u averages the values of $P@n$ over the positions n of the relevant tweets. Finally, the MAP value is computed as the mean of AP_u over the set of all users.

In addition, we used the Normalized Discount Cumulative Gain (NDCG) to evaluate our framework. It considers relevance of returned tweets in the resulting list and is calculated, for each user, as following:

$$NDCG_u@n = Z_n \sum_{i=1}^{n} \frac{rel(i)}{\log_2(i+1)} \tag{4}$$

where n is the evaluated position, $rel(i)$ is the analogous of $rel(n)$ in the previous equations, and Z_n is a normalization factor given by an ideal ordering (i.e., all $rel(i)$ with value 1). Later, in this article, we will refer to $NDCG@n$ as the mean of $NDCG_u@n$ over the set of all users.

5.3. Experimental Results

Starting from the test set described in Section 5.1, the ranking algorithm compares between them all tweets in the list, exploiting the deep model constructed during training execution. In particular, regarding comparison between the tweets, we considered a fixed day and variable time slots (see Section 4). In this way, we intend to evaluate the affinity between the daily interval and the user's re-tweet tendencies. At the end of execution, we compare the list of ranked tweets with the set of re-tweeted/replied tweets and use the MAP and NDCG measures to evaluate the results.

The results in terms of $P@n$ are shown in Fig. 4, while in terms of $MAP@10$ and $NDCG@10$ are in Fig. 5. Both figures highlight a very good precision in the first and last time slot (i.e., $Morning$ and $Night$), while a discrete difficulty in the other two (especially in the $Evening$ slot). Such result is probably due to an undersized training set in terms of contained week-ends: the deep model can not accurately identify user's interests in all slots since his behavior can change a lot in holidays against weekdays.

Furthermore, the experimental results of the proposed approach have been compared with the following ranking criteria:

- Reverse Chronological Order (RCO): Tweets list is ranked in reverse chronological order (from the most recent to the oldest one).

- User's Interests Score (UIS): Tweets are ordered according to a score calculated by multiplying the characteristic vector corresponding to the tweet content and vector of frequencies of topics representing the level of interest for a user with respect to the fixed set of categories (see Section 4.1).

The main differences with respect to the proposed approach, named Time-aware Adaptive Tweet Rank (TATR), consist of the following aspects: we don't use a unique ranking criteria for all users, the ranking is based on user's interests and the moment at which some of them may be more relevant for him. In this sense, TATR inherits the advantages of a timed representation of user's profile.

In order to validate selected features and the role of the time, we also evaluate the performance with some different configurations of TATR. In particular, we add three tests:

- TATR − Publisher's Authority features (TATR - PA): the deep model has been trained with a copy of the original dataset whose are removed features relative to publisher's authority.

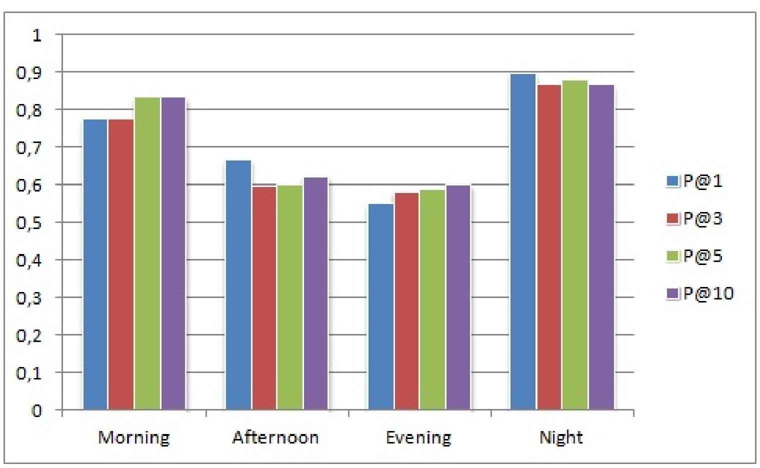

Figure 4: Precision of our approach measured in terms of $P@n$ in the different daily slots.

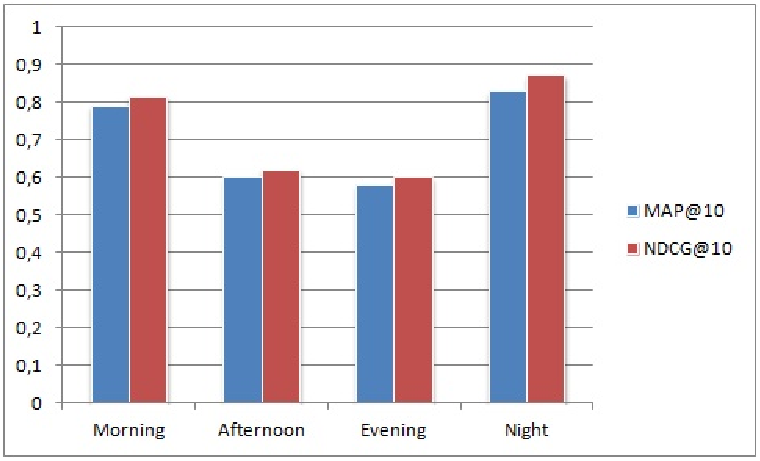

Figure 5: Precision of our approach measured in terms of MAP@10 and $NDCG$@10.

- TATR – Social Relation features (TATR - SR): the deep model has been trained with a copy of the original dataset, fewer features relative to the social relation between the user and the author of the tweet.
- TATR – Time-aware features (TATR - T): the deep model has been trained without information about time.

Summary of the performance is shown in Fig. 6. The experimental results reveal two aspects: the importance of time-awareness as feature aiming to adapt the ranking model for each user, and the influence of some features on the neural network. In fact, as shown in Fig. 6, values for RCO and UIS approaches turn out very low, and TATR has better performances also with respect to TATR-PA and TATR-SR in all considered slots, with values of MAP@10 that vary from 0.58 and 0.83. In particular, results show that among tweet evaluated features, "Publisher's Authority" (PA) ones have a better impact on performances. In fact, MAP@10 values substantially decrease when we omit PA features, while when we omit "Social Relation" (SR) features, the performance drop is less important. This should mean that users give more importance to the author's reputation rather than social relation existing with the authors. Regarding time-awareness, instead, results show a tendency of TATR-T not only to have worse

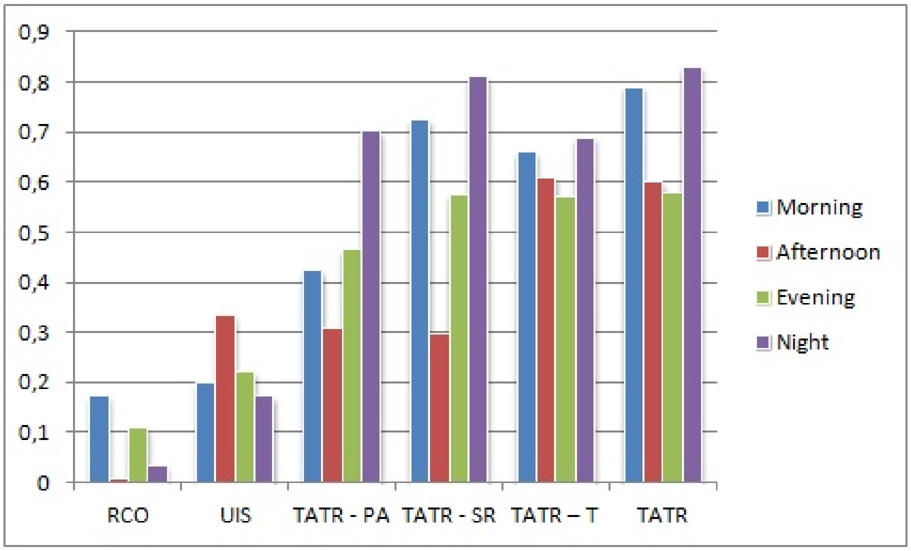

Figure 6: Values of MAP@10 of different approaches in the daily slots.

performances with respect to TATR but also almost constant values in different slots, highlighting the importance of setting time features in a recommender system of this kind.

In the nutshell, the experimental results reveal that by using both semantic (i.e., user's interest and tweet content) and time-awareness allow us to carry out promising results.

5.4. Training efficiency

We tested the framework by considering different features sets in order to uncover their contribution on the tweet ranking results. It follows that we trained the neural network by varying the configuration (as also explained in Section 2.2). Each configuration corresponding to the different set of features (i.e., inputs to the network), in turn, exhibited different training time, see Figure 7.

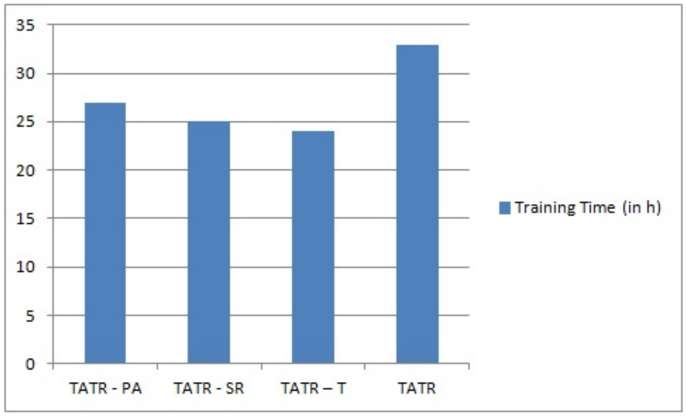

Figure 7: Training times for different network configurations.

6. Conclusion and Future Works

This work proposed personalized, adaptive and time-aware tweet ranking scheme implementing a learning to rank algorithm by means of a deep neural network. The ranking model is time-aware because among others the system foresees as input features also datatime corresponding to the re-tweets, or replies in order to achieve better performance when the interest of the user change along the timeline. The adaptivity is achieved by implementing the continuous training over the incoming tweet stream. We use a pairwise approach assuming that the user preference for a tweet is expressed when he/she posts a re-tweets, or reply ignoring their temporal order.

We evaluate a ranking model by measuring how many tweets that will be re-tweeted are included in the top-ranked tweet list when re-tweet is posted by the user. The results reveal promising performances and we point out that the most impacting features are: publisher authority, tweet content measures, and time-awareness.

Future works will move towards the following aspects. Since most Twitter users rarely post tweets of their own [10], there is a need to estimate the person interests considering their *friends* on Twitter. At the moment, we consider the user previous interests by taking into account the topics extract from his tweets and re-tweets, in the future social relations may be used to infer other user interests, in particular we may use graded logics to this aim using the approach of [47, 48]. In addition, we would like to experiment a novel advancements in the area of deep learning applied to text using word embeddings (i.e., *word2vec* [49]) method to represent tweet content. Finally, we would like to extend the proposed ranking scheme to face with misinformation problem, that means identifying those features that allow the system to produce a ranking model that top-ranks the most *qualitative* tweets.

References

[1] A. Kamilaris, G. Taliadoros, A. Pitsillides, D. Papadiomidous, The practice of online social networking of the physical world, International Journal of Space-Based and Situated Computing 2 (4) (2012) 240–252.

[2] C. Chen, S. Wen, J. Zhang, Y. Xiang, J. Oliver, A. Alelaiwi, M. M. Hassan, Investigating the deceptive information in twitter spam, Future Generation Computer Systems.

[3] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, Y. Yu, Collaborative personalized tweet recommendation, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2012, pp. 661–670.

[4] D. H. Alahmadi, X.-J. Zeng, Twitter-based recommender system to address cold-start: A genetic algorithm based trust modelling and probabilistic sentiment analysis, in: Tools with Artificial Intelligence (ICTAI), 2015 IEEE 27th International Conference on, IEEE, 2015, pp. 1045–1052.

[5] Y. Zhao, S. Liang, J. Ma, Personalized re-ranking of tweets, in: International Conference on Web Information Systems Engineering, Springer, 2016, pp. 70–84.

[6] C. De Maio, G. Fenza, V. Loia, M. Parente, Time aware knowledge extraction for microblog summarization on twitter, Information Fusion 28 (2016) 60–74.

[7] C. De Maio, G. Fenza, V. Loia, M. Parente, Online query-focused twitter summarizer through fuzzy lattice, in: 2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), 2015, pp. 1–8. `doi:10.1109/FUZZ-IEEE.2015.7337927`.

[8] G. De Francisci Morales, A. Gionis, C. Lucchese, From chatter to headlines: harnessing the real-time web for personalized news recommendation, in: Proceedings of the fifth ACM international conference on Web search and data mining, ACM, 2012, pp. 153–162.

[9] C. De Maio, G. Fenza, V. Loia, F. Orciuoli, Unfolding social content evolution along time and semantics, Future Generation Computer Systems 66 (2017) 146–159.

[10] M. Pennacchiotti, F. Silvestri, H. Vahabi, R. Venturini, Making your interests follow you on twitter, in: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM, 2012, pp. 165–174.

[11] M. Ye, X. Liu, W.-C. Lee, Exploring social influence for recommendation: a generative model approach, in: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, ACM, 2012, pp. 671–680.

[12] Y. Duan, L. Jiang, T. Qin, M. Zhou, H.-Y. Shum, An empirical study on learning to rank of tweets, in: Proceedings of the 23rd International Conference on Computational Linguistics, COLING '10, Association for Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 295–303.

[13] T.-Y. Liu, et al., Learning to rank for information retrieval, Foundations and Trends® in Information Retrieval 3 (3) (2009) 225–331.

[14] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, G. Hullender, Learning to rank using gradient descent, in: Proceedings of the 22nd international conference on Machine learning, ACM, 2005, pp. 89–96.

[15] Y. Song, H. Wang, X. He, Adapting deep ranknet for personalized search, in: Proceedings of the 7th ACM international conference on Web search and data mining, ACM, 2014, pp. 83–92.

[16] L. Rigutini, T. Papini, M. Maggini, F. Scarselli, Sortnet: Learning to rank by a neural preference function, IEEE Transactions on Neural Networks 22 (9) (2011) 1368–1380.

[17] D. F. Gurini, F. Gasparetti, A. Micarelli, G. Sansonetti, Temporal people-to-people recommendation on social networks with sentiment-based matrix factorization, Future Generation Computer Systems.

[18] H. Chen, X. Cui, H. Jin, Top-k followee recommendation over microblogging systems by exploiting diverse information sources, Future Generation Computer Systems 55 (2016) 534–543.

[19] T.-Y. Hsu, A. D. Kshemkalyani, Variable social vector clocks for exploring user interactions in social communication networks, International Journal of Space-Based and Situated Computing 5 (1) (2015) 39–52.

[20] D. Namiot, M. Sneps-Sneppe, Social streams based on network proximity, International Journal of Space-Based and Situated Computing 3 (4) (2013) 234–242.

[21] F. Zhao, Y. Zhu, H. Jin, L. T. Yang, A personalized hashtag recommendation approach using lda-based topic model in microblog environment, Future Generation Computer Systems 65 (2016) 196–206.

[22] X. Wan, J. Xiao, Collabrank: towards a collaborative approach to single-document keyphrase extraction, in: Proceedings of the 22nd International

Conference on Computational Linguistics-Volume 1, Association for Computational Linguistics, 2008, pp. 969–976.

[23] X. Wan, J. Xiao, Single document keyphrase extraction using neighborhood knowledge., in: AAAI, Vol. 8, 2008, pp. 855–860.

[24] X. Hu, X. Zhang, C. Lu, E. K. Park, X. Zhou, Exploiting wikipedia as external knowledge for document clustering, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 389–396.

[25] C. De Maio, G. Fenza, M. Gallo, V. Loia, M. Parente, Social media marketing through time-aware collaborative filtering, Concurrency and Computation: Practice and Experience, Wiley Online Library, 2017, In press.

[26] Y.-D. Seo, Y.-G. Kim, E. Lee, D.-K. Baik, Personalized recommender system based on friendship strength in social network services, Expert Systems with Applications 69 (2017) 135–148.

[27] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, H. Zha, Like like alike: joint friendship and interest propagation in social networks, in: Proceedings of the 20th international conference on World wide web, ACM, 2011, pp. 537–546.

[28] M. Islam, C. Ding, C.-H. Chi, Personalized recommender system on whom to follow in twitter, in: Big Data and Cloud Computing (BdCloud), 2014 IEEE Fourth International Conference on, IEEE, 2014, pp. 326–333.

[29] Y. Hao, Y. Zhang, J. Cao, Web services discovery and rank: An information retrieval approach, Future Generation Computer Systems 26 (8) (2010) 1053–1062.

[30] X. Li, H. Wang, B. Ding, X. Li, D. Feng, Resource allocation with multi-factor node ranking in data center networks, Future Generation Computer Systems 32 (2014) 1–12.

[31] A. Dessi, M. Atzori, A machine-learning approach to ranking rdf properties, Future Generation Computer Systems 54 (2016) 366–377.

[32] D. Chen, J. Yan, G. Wang, Y. Xiong, W. Fan, Z. Chen, Transrank: A novel algorithm for transfer of rank learning, in: 2008 IEEE International Conference on Data Mining Workshops, IEEE, 2008, pp. 106–115.

[33] D. Chen, Y. Xiong, J. Yan, G.-R. Xue, G. Wang, Z. Chen, Knowledge transfer for cross domain learning to rank, Information Retrieval 13 (3) (2010) 236–253.

[34] P. Geurts, G. Louppe, Learning to rank with extremely randomized trees., in: Yahoo! Learning to Rank Challenge, 2011, pp. 49–61.

[35] B. K. Li Deng, Geoffrey Hinton, New types of deep neural network learning for speech recognition and related applications: An overview, in: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2013, 2013.

[36] D. C. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Mitosis detection in breast cancer histology images with deep neural networks, in: International Conference on Medical Image Computing and Computer-assisted Intervention, Springer, 2013, pp. 411–418.

[37] M. Kågebäck, Deep learning for nlp.

[38] P. Mirowski, M. Ranzato, Y. LeCun, Dynamic auto-encoders for semantic indexing, in: Proceedings of the NIPS 2010 Workshop on Deep Learning, 2010, pp. 1–9.

[39] D. Tang, F. Wei, B. Qin, T. Liu, M. Zhou, Coooolll: A deep learning system for twitter sentiment classification, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 208–212.

[40] G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.

[41] H. Kwak, C. Lee, H. Park, S. Moon, What is Twitter, a social network or a news media?, in: WWW '10: Proceedings of the 19th international conference on World wide web, ACM, New York, NY, USA, 2010, pp. 591–600. `doi:http://doi.acm.org/10.1145/1772690.1772751`.

[42] R. Mihalcea, A. Csomai, Wikify!: linking documents to encyclopedic knowledge, in: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management, ACM, 2007, pp. 233–242.

[43] J. Weng, E.-P. Lim, J. Jiang, Q. He, Twitterrank: finding topic-sensitive influential twitterers, in: Proceedings of the third ACM international conference on Web search and data mining, ACM, 2010, pp. 261–270.

[44] B. McFee, G. R. Lanckriet, Metric learning to rank, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 775–782.

[45] N. Craswell, Precision at n, in: Encyclopedia of database systems, Springer, 2009, pp. 2127–2128.

[46] K. Järvelin, J. Kekäläinen, Ir evaluation methods for retrieving highly relevant documents, in: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 2000, pp. 41–48.

[47] M. Faella, M. Napoli, M. Parente, Graded alternating-time temporal logic, in: Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers, 2010, pp. 192–211. `doi:10.1007/978-3-642-17511-4_12`.
URL `https://doi.org/10.1007/978-3-642-17511-4_12`

[48] A. Ferrante, M. Napoli, M. Parente, Graded-ctl: Satisfiability and symbolic model checking, in: Formal Methods and Software Engineering, 11th International Conference on Formal Engineering Methods, ICFEM 2009, Rio

de Janeiro, Brazil, December 9-12, 2009. Proceedings, 2009, pp. 306–325. doi:10.1007/978-3-642-10373-5_16.
URL https://doi.org/10.1007/978-3-642-10373-5_16

[49] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv:1301.3781.