Contextualizing Trending Entities in News Stories

Marco Ponza
Bloomberg L.P.
London, United Kingdom
mponza@bloomberg.net

Diego Ceccarelli
Bloomberg L.P.
London, United Kingdom
dceccarelli4@bloomberg.net

Paolo Ferragina
University of Pisa
Pisa, Italy
paolo.ferragina@unipi.it

Edgar Meij
Bloomberg L.P.
London, United Kingdom
emeij@bloomberg.net

Sambhav Kothari
Bloomberg L.P.
London, United Kingdom
skothari44@bloomberg.net

ABSTRACT

Trends are those keywords, phrases, or names that are mentioned most often on social media or in news in a particular timeframe. They are an effective way for human news readers to both discover and stay focused on the most relevant information of the day. In this work, we consider trends that correspond to an entity in a knowledge base and introduce a new and as-yet unexplored task of identifying other entities that may help explain the "*why*" an entity is trending. We refer to these retrieved entities as *contextual* entities. Some of them are more important than others in the context of the trending entity and we thus determine a ranking of the contextual entities according to how useful they are in explaining the trend.

We propose two solutions for ranking contextual entities. The first one is fully unsupervised and based on Personalized PageRank, calculated over a trending entity-specific graph of other entities where the edges encode a notion of directional similarity based on embedded background knowledge. Our second method is based on learning to rank and combines the intuitions behind the unsupervised model with signals derived from hand-crafted features in a supervised setting. We compare our models on this novel task by using a new, purpose-built test collection created using crowdsourcing. Our methods improve over the strongest baseline in terms of Precision at 1 by 7% (unsupervised) and 13% (supervised). We find that the salience of a contextual entity and how coherent it is with respect to the news story are strong indicators of relevance in both unsupervised and supervised settings.

ACM Reference Format:
Marco Ponza, Diego Ceccarelli, Paolo Ferragina, Edgar Meij, and Sambhav Kothari. 2018. Contextualizing Trending Entities in News Stories. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/1122445.1122456

Woodstock '18, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

Every day, millions of news stories are produced and delivered to readers through newspapers, websites, and social media. Given such an overwhelming amount of data, it is important to provide automatic tools that can identify only the most relevant information, as well as help the users to understand *why* a particular piece of information was selected.

Trends are a way to distill units such as keywords, phrases, or names from news stories that are meaningful for characterizing the content. These units are important because they allow human news readers to discover and stay focused on the most relevant information of the day.

In this work, we focus on trends that correspond to a particular entity in a knowledge base (KB), called the *trending* entity, and we introduce the new and as-yet unexplored task of identifying other entities that help explain why it is trending. We refer to these retrieved entities as *contextual entities*. Some of them are more important than others in explaining the trending entity, and our primary motivation for our work to rank them accordingly.

Note that our work does not address detecting the trending entity per se (which we assume to be given) but rather focuses on the *retrieval and ranking* of contextual entities from the news stories that mention the trending one. For example, `Joe Biden` should be retrieved and ranked as a top contextual entity for the trend about the `2020 United States presidential election`.

The contextualization of trending entities is a fundamental problem that can help a variety of downstream applications. For example, the retrieved contextual entities might be used as pivot for creating entity-driven summaries [1, 4], or for providing an explicit context when the identification of the type of entity-bearing queries [10] is needed. Contextual entities might be also used to expand or help topical search queries [5, 7, 25].

To this end, we make the following contributions. (i) We propose an unsupervised graph-based algorithm that leverages Personalized PageRank and entity embeddings [12, 27]. We show that this method outperforms several baselines. In particular, it improves the strongest baseline based on entity salience by 7% for Precision at 1. (ii) We propose a supervised method based on learning to rank that combines multiple signals and achieves even further, significant improvements by 6% for Precision at 1. (iii) As no prior test collection for evaluating our task exists, we create and release a novel dataset using a crowdsourcing methodology.

2 RELATED WORK

The task of contextualizing entities has been tackled by several authors, e.g., through "explaining" a triple of two entities and a relationship or identifying domain-specific paths between entities [19, 22]. Also related to our setting is quantifying so-called relatedness, i.e., estimating the strength of the relationships between two or more entities, for instance by considering co-occurrences in a set of documents [16]. Similarly, entity salience aims at quantifying how crucial an entity is in natural language discourse [8, 18, 21, 24, 26].

Our problem differs from this earlier work in that we assume that the input is a set of documents coupled with a *trending* entity that is mentioned in them. Our goal is to retrieve a set of relevant, contextual entities that help explain why the query entity is trending. The earlier work that we presented above is intrinsically different as those assume one or more query entities and have as goal to identify other entities according to some notion of salience, while our goal is to find a set of entities that help to explain the query entity in the context of a set of documents, i.e., a trend. Because these approaches are related to our problem, we use them as baselines below.

3 PROBLEM STATEMENT

We assume to work on a stream of news stories $\mathcal{S}$ and we use entity linking to map each news story to a set of entities from a KB. In this work, we consider Wikipedia as our KB and each Wikipedia article constitutes an entity. Our stream of news stories continuously evolves, so we refer to $\mathcal{S}$ as the subset of news stories published in a particular timeframe, e.g., a day, and to $\mathcal{S}_e \subseteq \mathcal{S}$ as the subset of stories about the entity e. Similarly, we use $\mathcal{S}_{e_i,e_j} \subseteq \mathcal{S}$ to denote the subset of stories where both e_i and e_j occur. A trend $\mathcal{T}_{e_t} = \langle e_t, \mathcal{S}_{e_t} \rangle$ consists of a trending entity e_t and a set of stories $\mathcal{S}_{e_t}$. Given a trend $\mathcal{T}_{e_t}$, the set of contextual entities $\mathcal{E}_{\mathcal{T}_{e_t}} = \{e_c \mid e_t \neq e_c \text{ and } \mathcal{S}_{e_t,e_c} \neq \emptyset\}$ consists of those entities co-occurring with e_t in at least one news story in $\mathcal{S}$.

Task Definition

Given an ideal ranking function $\sigma : \langle \mathcal{T}_{e_t}, \mathcal{E}_{\mathcal{T}}, e_c \rangle \mapsto \mathbb{R}$ that assigns a ranking score to a contextual entity $e_c \in \mathcal{E}_{\mathcal{T}_{e_t}}$ according to its *usefulness in explaining* why the given trending entity e_t is actually trending, the goal is to find a function $\widetilde{\sigma}$ that best approximates the ideal function σ.

In the following, we show how we build the ideal ranking function σ using human annotators (Section 5.2), and we explore unsupervised (Section 4.1) and supervised (Section 4.2) solutions for finding a good approximation $\widetilde{\sigma}$. The quality of $\widetilde{\sigma}$ is evaluated against σ by using standard IR metrics that we define in Section 6.2.

4 RANKING CONTEXTUAL ENTITIES

We implement $\widetilde{\sigma}$ by proposing two main methods. Both take as input the trend $\mathcal{T}_{e_t}$ and the contextual entities $e_c \in \mathcal{E}_{\mathcal{T}_{e_t}}$ and then return a score for each e_c.

The first solution solves the task using an unsupervised approach which aims at scoring each contextual entity by executing Personalized PageRank [12] on a specifically designed graph of entities built from $\{e_t\} \cup \mathcal{E}_{\mathcal{T}_{e_t}}$ through the use of entity embeddings for the weighting of its edges [27]. We hypothesize that we can improve over this unsupervised method by incorporating hand-crafted features. Our second solution is therefore supervised and ranks the contextual entities in $\mathcal{E}_{\mathcal{T}_{e_t}}$ according to a score obtained using learning to rank. Both approaches are detailed below.

4.1 Personalized PageRank with Embeddings

Our first and unsupervised method works in three stages. First, a weighted directed graph is built by mapping contextual entities to nodes and entity relations to edges. Edges are weighted by deploying entity embeddings [27]. Second, the teleport vector of the Personalized PageRank algorithm (PPR in short) is instantiated. Third, PPR is executed and its computed scores are used as estimators of the goodness of contextual entities.

Stage 1: Construction. Nodes of our graph are the entities in $\{e_t\} \cup \mathcal{E}_{\mathcal{T}_{e_t}}$. Edges are built in two different ways. The first set of edges is drawn in order to pivot the graph relationships around the trending entity e_t, which is known to be central for the input trend. Each contextual entity $e_c \in \mathcal{E}_{\mathcal{T}_{e_t}}$ is thus connected with a direct edge to e_t, and vice versa. Edges are directed because in Stage 2 we weight them according to their semantic similarity that produces a different weight based on the direction of the edge. Additionally, a second set of edges is drawn in order to connect each contextual entity with other contextual entities that share the same topic across the news stories of $\mathcal{T}_{e_t}$. Each contextual entity e_{c_i} is thus connected with a direct edge to another contextual entity e_{c_j} if they co-occur together in at least one news story of $\mathcal{T}_{e_t}$.

Stage 2: Weighting. Given the set of edges from the previous stage, we strengthen or weaken every edge with respect to how much two entities are semantically similar. The underlying idea is to use a weighting scheme that enables us to introduce a sense of *background knowledge* derived from the KB. An edge from entity e_i to entity e_j is weighted by computing the cosine similarity between the embedding vectors of the two entities [27] and then normalizing the result with respect to the sum of the weights of the edges outgoing from e_i, namely:

$$weight(e_i, e_j, \tau) = \frac{weight'(e_i, e_j, \tau)}{\sum_{e_k \in Out(e_i)} weight'(e_i, e_k, \tau)}$$

$$weight'(e_i, e_j, \tau) = \begin{cases} \text{cosine}(W_{e_i}, W_{e_j}) & \text{if cosine}(W_{e_i}, W_{e_j}) \geq \tau \\ 0 & \text{otherwise,} \end{cases}$$

where $Out(e_i)$ is the set of entities connected by a directed edge from e_i; W_e is the embedding vector of the entity e; and $\tau = 0.3$ is used to remove noisy edges' weights with low similarity scores [27].

Stage 3: Teleport. The last element that needs to be defined before executing PPR, is the instantiation of the teleport vector p (i.e., a vector that encodes a prior importance score for each entity in the graph), calculated as $p(e_i) = \frac{p'(e_i)}{\|p'\|}$, where

$$p'(e_i) = \begin{cases} 1 & \text{if } i = t \\ score'(e_i, \mathcal{T}_{e_t}) & \text{otherwise} \end{cases} \quad (1)$$

and with $score'$ is a function we will define later in the experiments (see Section 6.1 and Table 1). The intuition here is that the teleport

vector allows us to introduce more signals such as relative position or salience into the ranking algorithm–through the *score′* function.

Stage 4: Ranking. The ranking score for every contextual entity e_{c_i} is given by executing PPR on our graph of contextual entities for k steps:

$$rank^{(k)}(e_{c_i}) = (1-d)\ p(e_{c_i}) + d\left(\sum_{e_j \in Out(e_{c_i})} rank^{(k-1)}(e_j) \cdot weight(e_j, e_{c_i})\right)$$

where $d \in (0,1)$ is the so called PPR's damping factor. In our experiments, we use the classic setting of $d = 0.85$ [2].

4.2 Learning to Rank Entities

Our second, supervised method works in two stages. Given a trend $\mathcal{T}_{e_t}$, each contextual entity $e_c \in \mathcal{E}_{\mathcal{T}_{e_t}}$ is transformed into a vector of engineered features, and then, these vectors are ranked by a learning to rank model (LTR in short).

Stage 1: Feature Engineering. We group features according to the "source of information" used to generate them. Below we list the sources we used.

• Position. This set of features aims to provide a sense on *where* a contextual entity e_c is mentioned in the news stories of the trend $\mathcal{T}_{e_t}$. Since important information is usually mentioned at the beginning of a news story, this set of features should be able to properly identify the correct pattern of relevant contextual entities based on positional signals. Specifically, we calculate the average, minimum, maximum, and standard deviation of the character-offset positions where a contextual entity e_c occurs in the news stories $\mathcal{S}_{e_t,e_c}$.

• Frequency. These features model *how often* a contextual entity e_c is mentioned among the news stories of the input trend $\mathcal{T}_{e_t}$. The intuition is that the more it occurs the more it should be relevant as contextual entity for e_t. We compute the average, minimum, maximum, and standard deviation of the frequency of the entity e_c among the news stories $\mathcal{S}_{e_t,e_c}$.

• Co-Occurrence. In a similar fashion, entities that often co-occur together with the trending entity e_t within the same sentence should be more relevant than the ones co-occurring less often. We compute the average, minimum, maximum, and standard deviation of the co-occurrence of e_c with e_t in $\mathcal{S}_{e_t,e_c}$, as well as the number of stories where e_c appears.

• Popularity. These features estimate the *general popularity* of a contextual entity e_c measured through different statistics derived from the Wikipedia KB. The key idea is that often popular entities in Wikipedia (e.g., `Finance`, `Market`, ...) should receive a proper score that enables the machine learning model to filter them out. So we compute the *inverse document frequency* of e_c over the whole Wikipedia corpus as $\log(N/|In(e_c)|)$, where N and $In(e)$ are, respectively, the total number of pages and the number of pages hyperlinked to e (i.e., the in-degree of the node). We also consider the $|Out(e)|$ as the number of pages pointed to by e. These *degrees* are introduced as features.

• Text Coherence. This set of features models the *coherence* between each contextual entity e_c and the textual content of the news stories belonging to the input trend $\mathcal{T}_{e_t}$. Please note that a contextual entity e_c can be mentioned in different ways across several news stories. For example, in the text *"Obama is flying abroad. Barack loves travelling."*, an entity linker disambiguates the two mentions *"Obama"* and *"Barack"* to the same entity `Barack Obama`. For each mention m disambiguated with the entity e_c, we introduce as features the average, minimum, maximum, and standard deviation of its prior probability and TagMe's rho score [9]. The *prior* probability is the probability that m appears in Wikipedia referencing the page e_c. We also use as features a combination of the prior probabilities with multiple Milne&Witten relatedness [23] scores between e_c and a small subset of entities surrounding e_c's occurrences in the news stories of $\mathcal{T}_{e_t}$.

• Neural Coherence. This set of features attempts to provide a sense of *coherence* for a contextual entity e_c with respect to all the other entities extracted in the input trend $\mathcal{T}_{e_t}$. The first feature of this set is calculated as the cosine similarity between the embeddings of the entities e_c and e_t, namely $\text{cosine}(W_{e_c}, W_{e_t})$. The other features of this set are computed as average, minimum, maximum, and standard deviation of the cosine similarities between e_c and all the other contextual entities in $\mathcal{E}_{\mathcal{T}_{e_t}}$.

• Salience. This set of features estimates the *local importance* of a contextual entity e_c within a single news story of the input trend $\mathcal{T}_{e_t}$. A salience score is computed for every e_c in each trend's story by using the system SWAT [18]. Other features are computed as average, minimum, maximum, and standard deviation of entity salience scores. We also introduce two more sophisticated features. The first one implements the cosine similarity between the distributions of salience scores of e_c and e_t in the news stories $\mathcal{S}_{e_t}$ of the input trend $\mathcal{T}_{e_t}$. The second one implements the fraction of news stories, published earlier than the trend's day, where entities e_c and e_t occur and got a salience score greater then a fixed threshold.[1]

Stage 2: Ranking. For ranking the set of contextual entities given the above set of features, we can use any learning to rank framework available in the literature, or design a specialized neural network architecture for solving the task. For efficiency reasons, we decided to limit our choices on a popular and publicly available gradient boosting library. We chose the implementation provided by LightGBM [13], since it has been shown to achieve performance equal to or higher than XGBoost [3], but with significant improvements in both time and space.

5 CREATING A DATASET

In this section, we outline our procedure for building the test collection we use to evaluate our task of explaining trending entities.

We consider the New York Times Annotated Corpus [20] as our first stream of news stories. It is a publicly available dataset consisting of 1.8M news stories published by the New York Times between 1987 and 2007. Each news story is composed of different fields. For our experiments, we rely only on the fields *category*, *publication date*, *title*, *abstract*, and *body*. We focus on the 84K news stories belonging to the category "Business", since it is the primary source of information commonly used for the understanding and analysis of financial markets [6].

We then build our dataset via a two-stage, manual annotation process that is described in the following sections.

[1] We use a value of 0.4 in order to exclude entities with a low saliency score.

5.1 Stage 1: Generating Trending Entities

For the detection of trending entities by human annotators we proceed in two phases.

Phase 1: Extracting Candidate Trending Entities. We identify the set of candidate trending entities by first looking at entities from title and abstract of each news story in $\mathcal{S}$ by using the TagMe entity linking system [9]. This process produces a total of 1.91M entities with 148K unique ones. We then filter out rare candidate trending entities by discarding the ones that occur in less than two stories. Finally, we group the remaining entities per day, thus generating a set of candidate trends having each one the form $\mathcal{T}_{e_t} = \langle e_t, \mathcal{S}_{e_t} \rangle$. This filtering step produces a total of 924 candidate trends with a total of 3K news stories (4.2 per day and 3.88 per trend), spanning across 12 years of New York Times publishing (from 1996 to 2007).

Although trends can last more than a day in this work we decided to work on a day granularity identifying meaningful trends. We observe that using the day granularity might also simplify the annotation work, because a trend will always have news stories from a specific day.

Please note that we have also implemented and evaluated other automatic techniques to detect trending entities, such as the ones by Koike et al. [14] and by Guzman and Poblete [11]. In both these cases, we found that the two methods produced several false positive trending entities and in some cases missed the true positive ones. We attribute this behavior to the fact that the dataset comes from a single source of news stories (i.e., NYT), and that sometimes there are not many news stories referring to the same trending topic. That is why we decided to use the simple heuristic described above and then to rely on human annotators to filter out the correct ones (given also that detecting trends is not the focus of this work).

Phase 2: Manually Annotating Trending Entities. We ask eight human annotators to evaluate, whether trending or not, a random subset of 400 entities of the candidate trends generated from the previous phase. We consider this amount of entities enough for an annotator to stay focused over their evaluation. Before starting the human annotation process, we provide the annotators with a document formally describing the guidelines[2] for the annotation of candidate trending entities as actual trending or not.

During the annotation process, the human annotator is able to examine one trend $\mathcal{T}_{e_t} = \langle e_t, \mathcal{S}_{e_t} \rangle$ at a time. The annotator can click on the Wikipedia page of the entity e_t, as well as look at the titles and abstracts of trend's news stories $\mathcal{S}_{e_t}$. At the end of the annotation process, we have produced 202 actual trending entities with a total of 731 news stories.

5.2 Stage 2: Generating Relevant Contextual Entities

For the detection of contextual entities we proceed in two phases.

Phase 1: Extracting Contextual Entities. We run TagMe [9] on the bodies of news stories of the 202 trends generated from the previous stage (Section 5.1) in order to extract candidate contextual entities.

[2]Details can be found at ZENODOURL

Phase 2: Manually Annotating Contextual Entities. We ask eight human annotators to assign a relevance score to each candidate contextual entity according to how relevant it is for explaining *why* the entity e_t is trending for the trend $\mathcal{T}_{e_t} = \langle e_t, \mathcal{S}_{e_t} \rangle$.

To simplify the task, we show to the annotators only the contextual entities in the body that are also mentioned in the title or in the abstract of the news in $\mathcal{S}_{e_t}$. The intuition is that it is unlikely that an entity that is relevant for the story and the trend is not mentioned in the title or in the abstract. This also allows us to speed up the whole human annotation process (a similar technique has been used in the past by Dunietz and Gillick [8]).

The annotation interface for this phase is similar to the previous one but contains also the candidate contextual entities, and it allows the human annotator to click on the Wikipedia page of both the trending and the contextual entities. Before starting this second human annotation process, we provide annotators with a document formally describing the guidelines[2] for the labelling of contextual entities with three different relevance scores.

After the human annotation process, we remove 49 trends whose contextual entities are annotated with only "Irrelevant" labels by at least two human annotators. On the resulting 153 trends, we measure the agreement with different correlation coefficients, resulting in a Kendal's τ score of 0.603, Cohen's kappa of 0.508, and Fleiss's kappa of 0.439. These values are respectively considered good and moderate, as well as consistent with those reported in works that involve similar human annotation tasks [21, 22].

Finally, we aggregate the different annotated relevance scores by their average among annotators. The resulting scores associated with the contextual entities are then mapped into a new set of relevance scores depending on their mean: "Relevant" (score = 2), "Somewhat Relevant" (score = 1), and "Irrelevant" (score = 0), when the averaged score is in $[1.5, 2.0]$, $(0.5, 1.5)$, and $[0, 0.5]$, respectively

6 EXPERIMENTAL SETUP

In this section, we describe the setup of our experimental evaluation. We provide implementation specifics as well as baselines and metrics used for the evaluation.

The first part of our experimental evaluation focuses on the unsupervised approach. We ask:

RQ1. How does our unsupervised method perform compared to baselines?

RQ2. How does it qualitatively differ from the best baseline?

We answer these questions by experimenting with our unsupervised solution based on PPR and compare it against several baselines on the *complete dataset* consisting of 149 trends.

The second part of our experimental evaluation addresses the following research questions:

RQ3. How does our supervised approach (LTR) perform compared to PPR and the best baseline?

RQ4. How does LTR qualitatively differ from PPR?

RQ5. How do PPR, LTR, and the best baseline perform with respect to trends when we have highly ranked contextual entities that exhibit low salience?

In order to evaluate LTR we split the dataset into a training set (consisting of 50 trends with 36K entities from 1996 to 2000), a development set (consisting of 34 trends with 26K entities from 2000 to 2001), and a test set (consisting of 65 trends with 57K entities from 2002 to 2007). We train LightGBM [13] by optimizing its hyper-parameters (over training/development sets) through a grid-search in which the parameters range over n_estimators $\in [10^2, 10^3]$ and max_depth $\in [2, 10]$.[3] The best performing model on the development set is then applied to the test set.

Implementation Specifics. Our experiments rely on the English Wikipedia dump of November 2019. We use Wikipedia2Vec [27] for processing Wikipedia and generating the Wikipedia graph from its hyperlink structure. We also use the same tool for the generation of entities embeddings from Wikipedia. All embeddings are learned with their default configurations and vectors' size of 100. Entity salience scores are calculated with SWAT [18], a state-of-the-art and publicly available entity salience system.

6.1 Baselines

To the best of our knowledge, there is no previously published method for addressing the new research task introduced in this paper. Hence, we design several baselines that are used to show the efficacy challenges underlying this task. Some of these baselines will be also used to instantiate the teleport scoring function $score'$ in Equation 1 of our unsupervised solution PPR. Table 1 reports the baselines we use in the experiments.

Please note that the baselines and the two solutions we present in Section 4 do not use any signal derived from title and abstract of the news stories. The motivation behind this choice relies on the fact that titles and abstracts are used to extract the set of entities provided to the human annotators (see Section 5.2, Phase 2). A similar methodology was adopted by Dunietz and Gillick [8] in the context of entity salience.

Because the baselines hinge upon the same signals on which we also designed the features of our LTR solution (Section 4.2), we distinguish between baselines methodologies and the group of features by using the normal or the SMALL CAPS fonts, respectively. For example, Salience will refer to one baseline methodology, whereas SALIENCE will indicate the group of features based on salience signals.

6.2 Evaluation Metrics

We use standard ranking evaluation metrics: Mean Average Precision (MAP), Precision at k (P@k, with $k \in \{1, 2, 3\}$), Normalized Discount Cumulative Gain at k (NDCG@k, with $k \in \{5, 10\}$), and Mean Reciprocal Rank (MRR). We select these metrics because we imagine a scenario where we show to the user from one to five related contextual entities ranked by relevance to explain a trend. MAP, MRR, and P@k expect binary labels and we define "Relevant" and "Somewhat Relevant" labels as positive. For NDCG, we use a three-point scale of relevance scores: "Relevant" (score = 2), "Somewhat Relevant" (score = 1), and "Irrelevant" (score = 0). These

[3]We also tried to tune these hyper-parameters using larger/lower values, as well as tuning them together with other hyper-parameters with no differences.

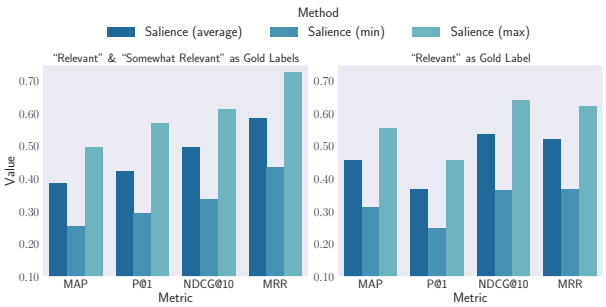

Figure 1: Results of Salience baseline deploying three different aggregation functions (average, min, and max).

results are reported in the left-most part of tables and figures under the header *"Relevant" & "Somewhat Relevant" as Gold Labels*.

In order to provide insight into the performance of methods using the highest human annotated relevance scores, we also calculate the same metrics by considering "Somewhat Relevant" entities as "Irrelevant". These second set of results are reported in tables and figures under the header *"Relevant" as Only Gold Label*.

In all the experiments, statistically significant differences are calculated with a Student's paired two-tailed t-test.

7 RESULTS AND DISCUSSION

7.1 Unsupervised Evaluation

In the first part of this section we focus on PPR and Salience across different configurations of the teleport vector and aggregation function, respectively. In the second part, their best performing configurations are used to answer both **RQ1** and **RQ2**.

Figure 1 shows the results of Salience baseline instantiated with different aggregation functions. Maximum outperforms all the other configurations among all metrics. This first experiment already shows that entity salience scores can be good indicators of pertinent contextual entities.

Figure 2 shows the results of PPR where the teleport function $score'$ of Equation 1 is instantiated through the use of different baselines. For the sake of comparison, we also report the method executed by instantiating the teleport vector with the uniform distribution (named Uniform in the plot). We also tried to instantiate the teleport vector with other baselines, but without achieving significant improvements with respect to the ones shown in the plots. Uniform function achieves the lowest performance, thus showing the importance (and the sensitivity) that the teleport vector has on to the overall ranking of entities. Not surprisingly, the best results are achieved by instantiating the teleport vector with a method based on entity salience.

We derive Table 2 in order to answer **RQ1** and compare our unsupervised solution against simple baselines. In the following, for Salience and PPR we always respectively report only the results achieved with their best configuration shown above in the plots (i.e., max for Salience and average for PPR). Almost all baselines result in low performance, with the only exception of Salience—which is actually based on a supervised entity salience system (i.e., SWAT [18]) specifically designed for the the detection of salient entities—being able to achieve satisfying results. The last row of

Table 1: Description of the baselines used in the experiments. The last two baselines are also used as functions for the instantiation of the PPR's teleport vector ($score'$)

Baseline	Description
Frequency	This method scores each contextual entity with respect to its frequency in the news stories for the given trend.
Position	This method scores each contextual entity with respect to the average of the inverse of its position over the news stories of the given trend, so that a high position score is assigned to entities at the beginning of the news stories.
Co-Occurrence	This method scores each contextual entity based on how many sentences in the given trend mention the contextual entity and the trending entity. This can be seen as a more granular version of *Frequency*, that looks at the co-occurrence within the news story rather than within a sentence.
PMI	This method scores each contextual entity with the pointwise mutual information (PMI) with respect to the trending entity at hand. PMI is here calculated at sentence-level among the news stories of the given trend.
Milne&Witten	This method scores each contextual entity with the Milne&Witten relatedness score [23] with respect to the trending entity. The Milne&Witten score is derived from the whole directed Wikipedia graph.
Jaccard	This method scores each contextual entity by the Jaccard similarity with the trending entity, calculated between the undirected neighbourhood sets (derived from the whole Wikipedia graph) of the contextual entity and the one of the trending entity.
Wikipedia Embeddings	This method scores each contextual entity with respect to the cosine similarity between its entity embedding vector and the one of the trending entity.
Stories Embeddings	This method implements the approach by Mohapatra et al. [16], which learns entity embeddings from news stories in a specific time range. Entities are then scored with respect to their cosine similarity with an input entity. In our context, the time range is a single day, the news stories are the ones of the given trend, and the input entity is the trending one to analyze.
Reciprocal Rank	This technique is often used for merging results provided by different ranking methods [15]. In our case, we sort the contextual entities by the scores provided by Position and Co-Occurrence baselines. Then, we assign as final score the reciprocal of the ranking position obtained in each of them. We also use this method for instantiating the PPR's teleport scoring function.
Salience	This method works in two stages. First, it computes the entity salience scores of each contextual entity among the news stories in the given trend. Second, different entity salience scores—one for each pair ⟨contextual entity, news story⟩— are aggregated into a single ranking score. In our experiments, we will use average, minimum, and maximum aggregation functions. We also use this method for instantiating the PPR's teleport scoring function.

Table 2: Performance of the unsupervised solutions on the whole dataset. Statistically significant improvements between PPR and Salience (max) are indicated with $^{\blacktriangle}$ for $p < 0.05$ and with $^{\triangle}$ for $p < 0.1$.

Method	"Relevant" & "Somewhat Relevant" as Gold Labels						"Relevant" as Gold Label					
	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR
Frequency	0.098	0.262	0.224	0.168	0.233	0.448	0.097	0.208	0.177	0.179	0.242	0.382
Position	0.237	0.195	0.152	0.237	0.319	0.354	0.247	0.114	0.105	0.249	0.331	0.274
Co-Occurrence	0.359	0.477	0.295	0.441	0.479	0.604	0.441	0.416	0.221	0.486	0.515	0.528
PMI	0.147	0.161	0.15	0.186	0.209	0.324	0.173	0.107	0.105	0.195	0.219	0.252
Milne&Witten	0.177	0.141	0.136	0.179	0.242	0.311	0.177	0.094	0.087	0.174	0.24	0.23
Jaccard	0.214	0.248	0.183	0.234	0.276	0.394	0.229	0.174	0.116	0.240	0.282	0.299
Wikipedia Embeddings	0.206	0.221	0.154	0.214	0.274	0.372	0.213	0.154	0.096	0.210	0.276	0.276
Stories Embeddings	0.210	0.208	0.161	0.238	0.287	0.373	0.237	0.148	0.110	0.253	0.299	0.295
Reciprocal Rank	0.418	0.523	0.291	0.460	0.508	0.630	0.488	0.430	0.219	0.501	0.542	0.541
Salience (max)	0.497	0.570	**0.394**	0.556	0.612	0.727	0.555	0.456	**0.286**	0.593	0.640	0.622
PPR	**0.519**	**0.644**	0.391	**0.586**	**0.637**	**0.773**$^{\triangle}$	**0.605**$^{\blacktriangle}$	**0.564**$^{\blacktriangle}$	0.282	**0.639**$^{\triangle}$	**0.678**$^{\triangle}$	**0.686**$^{\triangle}$

Table 3: Examples of where PPR improves (top part) and hurts (bottom part) results with respect to the Salience (max). Pertinent and non-pertinent top-ranked contextual entities are marked with ✔ and ✗, respectively.

Trending Entity	Trend Description	Top Ranked Contextual Entities	
		Salience (max)	PPR
Frank Quattrone	Frank Quattrone's trial involving e-mails sent to his collegues at Credit Suisse First Boston.	1. Criminal charge ✔ 2. Initial public offering ✗ 3. Jury ✗	1. Criminal charge ✔ 2. Trial ✔ 3. Credit Suisse First Boston ✔
Jean-Marie Messier	Jean-Marie Messier resigned as chief executive from Vivendi, Barry Diller will probably call the shots.	1. France ✗ 2. Vivendi ✔ 3. Barry Diller ✔	1. Barry Diller ✔ 2. Vivendi ✔ 3. Board of Directors ✔
JPMorgan Chase	JPMorgan Chase acquires Bank One Corporation, with Jamie Dimon becoming chief operating officer.	1. Jamie Dimon ✔ 2. Bank One Corporation ✔ 3. Chicago ✗ 4. Sanford I. Weill ✔	1. Bank One Corporation ✔ 2. Jamie Dimon ✔ 3. Sanford I. Weill ✔ 4. Mergers and acquisitions ✔
Enron	Lawsuit against Enron for investments' manipulations.	1. Arthur Andersen ✔ 2. Paul Volcker ✔ 3. Kenneth Lay ✔	1. Accounting ✗ 2. Arthur Andersen ✔ 3. Employment ✗
Tyco International	Dennis Kozlowski leaves Tyco International despite recent salary bonus and benefits.	1. Dennis Kozlowski ✔ 2. Executive compensation ✔ 3. Board of directors ✔	1. Share (finance) ✗ 2. Stock ✗ 3. Dennis Kozlowski ✔

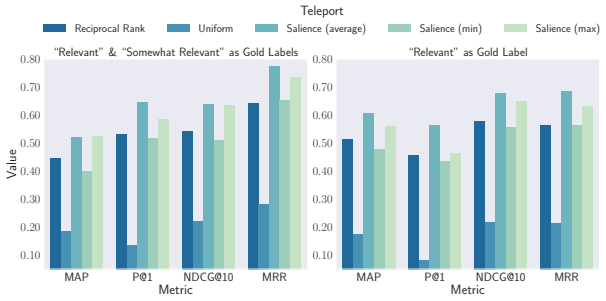

Figure 2: Results of our unsupervised solution based on PPR, considering different instantiations of the teleport vector.

Table 2 shows that our unsupervised solution based on PPR is able to bring several improvements with respect to the baseline Salience (max). We observe significant results among the metrics that measure the detection of the most relevant entities (right-most part of the table). Particular benefits are shown among MAP and P@1, which are here improved by 4.5% and 11% with respect to the Salience (max) baseline.

Table 3 reports several meaningful examples concerning where PPR qualitatively differs from Salience (max). PPR provides a better assignment of higher scores to the entities that are more central with respect to the trend's topics. PPR is more robust to the noise by properly filtering out contextual entities that have a high salience score in a single story, but they are actually not related with other trends' topics. Furthermore, Salience (max) usually ranks higher the geographic locations, despite they are rarely relevant contextual entities. This happens because the data used for training the entity salience system usually has locations labelled as salient entities in the ground-truth. These results allow us to conclude positively about **RQ2** and our PPR.

7.2 Supervised Evaluation

In this second set of experiments, we perform a quantitative and qualitative comparison of LTR, PPR, and the best baseline—i.e., Salience (max)—to answer questions **RQ3**, **RQ4**, and **RQ5**.

We derive Table 4 to answer **RQ3** and compare our supervised solution LTR against PPR and Salience (max) on the test set. This experiment shows that "Relevant" contextual entities can be easily identified with our unsupervised solution PPR. Its performance (right-most part of the table) is very close and not statistically significantly different from the ones achieved by LTR. On the other hand, when both "Relevant" and "Somewhat Relevant" entities need to be detected (left-most part of the table) PPR shows its limitations. LTR is here able to bring a larger number of advancements, with significant improvements against both Salience (max) and PPR across MAP, P@3, and NDCG@10 from 6.5% to 10.8%.

Table 5 reports several meaningful examples concerning where LTR qualitatively differs from PPR. As shown in the top half of the table, LTR results are more accurate, especially among the top ranked contextual entities. LTR partially solves the issue that affects PPR by avoiding to rank high "general" entities. Unfortunately for LTR—but for a fewer of cases than PPR—these "general" entities are still sometimes present in the results, but for a different reason: they are ranked higher because they appear at the very beginning of the trends' news stories, where usually relevant information is commonly introduced by the story's author. Because the pattern of relevant information present at the beginning of news stories occurs in most of the analysed examples, our features based on Popularity are not able to totally mitigate this phenomenon, thus leaving space in future work to the design of more sophisticated techniques for addressing this partially unsolved problem. These observations answer **RQ4**.

Because the best methods are based on salience signals, we decide to answer **RQ5** in their worst-case scenario. We analyze their performance on the subset of trends whose gold contextual entities have been wrongly associated with low salience scores. When this situation occurs, good contextual entities are harder to be properly detected: their salience here becomes misleading. Table 6 reports the results of our experiments, where only trends whose gold contextual entities achieved a salience score lower than the median (i.e., 0.7) are considered. This occurs on 75 trends of the complete dataset (first two rows of Table 6) and 33 trends of the test set (last two rows of Table 6). Here, PPR is more robust than both Salience (max) and LTR, with statistically significant improvements among more metrics than shown before. PPR—despite being unsupervised— improves the ranking provided by Salience (max), but also achieves similar or higher performance than LTR—which is supervised—when the salience scores do not fully represent how much a contextual entity is relevant among multiple stories.

8 MODEL SIMPLIFICATION

In the following, we perform an analysis of the features used in LTR, in order to see if it is possible to reduce the number of features used without compromising too much the performance metrics.

Feature Ablation. Figure 3 reports the performance of our solution where feature ablation is applied among every group of features. We observe that Salience and Text Coherence are the two groups of features that impact the most on the performance of our supervised solution: when the system does not use features based on the Salience group, performance suffers a significant loss, with a large decrease among all metrics of about 10%. This is somehow expected because in our previous experiments we showed that the best methods rely their predictions on salience scores. A similar behaviour can be also noticed for features based on Text Coherence. This is due to the fact that these features are beneficial for our system to filter out entities that are low related with the context where they have been extracted. On the other hand, Text Coherence features alone are not enough for reaching good performance (see Figure 4): a significant increase is possible only when also Salience features are employed. Figure 3 also shows that Neural Coherence features seem to slightly damage the system, decreasing its performance by 1% (not statistically significant).

Feature Selection. We now focus our analysis on selecting features from the two best performing groups detected above: namely, Text Coherence and Salience. Features are incrementally plugged in our solution until no statistically significant difference is detected between the model using all features and the one using only a subset of them. Table 7 shows the results of these experiments, where features are plugged in the model by their non-decreasing LightGBM

Table 4: Performance of different solutions on the test set. Significant improvements between LTR and PPR, resp. Salience (max), and are indicated with $^{\blacktriangle}$, resp. $^{\triangle}$, for $p < 0.05$.

Method	"Relevant" & "Somewhat Relevant" as Gold Labels						"Relevant" as Gold Label					
	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR
Salience (max)	0.474	0.569	0.364	0.526	0.584	0.714	0.534	0.462	0.251	0.566	0.616	0.604
PPR	0.495	0.646	0.364	0.565	0.617	0.767	0.591	0.554	0.256	0.622	0.659	0.665
LTR	**0.574**$^{\blacktriangle\triangle}$	**0.708**	**0.472**$^{\blacktriangle\triangle}$	**0.629**$^{\triangle}$	**0.682**$^{\blacktriangle\triangle}$	**0.815**$^{\triangle}$	**0.609**	**0.569**	**0.308**$^{\blacktriangle\triangle}$	**0.654**$^{\triangle}$	**0.696**$^{\triangle}$	**0.710**$^{\triangle}$

Table 5: Examples of where LTR improves (top part) and hurts (bottom part) results with respect to the PPR method. Pertinent and non-pertinent top-ranked contextual entities are marked with ✔ and ✗, respectively.

Trending Entity	Trend Description	Top Ranked Contextual Entities	
		PPR	LTR
United Airlines	United Airlines becomes under bankarupty protection and hires McKinse & Company for consultancy.	1. Airline ✗ 2. Customer ✗ 3. Employment ✗	1. McKinsey & Company ✔ 2. Bankruptcy ✔ 3. United States bankruptcy court ✔
DirecTV	Acquisition from News Corporation of the DirecTV's satellite system.	1. News Corporation ✔ 2. Rupert Murdoch ✔ 3. Stock ✗	1. News Corporation ✔ 2. Rupert Murdoch ✔ 3. Satellite television ✔
Federal Reserve	Federal Reserve leaves interests unchanged and different stock indices change their values.	1. Interest rate ✔ 2. Labour economics ✗ 3. Monetary policy ✗	1. Interest rate ✔ 2. Monetary policy ✗ 3. NASDAQ Composite ✔
Fannie Mae	Fannie Mae and Franklin Raines dispute on accounting irregularities that are currently investigated by Armando Falcon.	1. OFHEO ✔ 2. Franklin Raines ✔ 3. Armando Falcon ✔	1. OFHEO ✔ 2. Franklin Raines ✔ 3. Accounting ✗
CNOOC	CNOOC tries to buy Unlocal Corporation. Several financial institutions involved.	1. Unlocal Corporation ✔ 2. Goldman Sachs ✔ 3. Contract ✗	1. Unlocal Corporation ✔ 2. Presidency of George W. Bush ✗ 3. China ✗

Table 6: Performance by considering trends whose gold entities have salience score lower than the median. Statistically significant improvements between PPR and Salience (max) and LTR are indicated with $^{\blacktriangle}$ for $p < 0.05$ and with $^{\triangle}$ for $p < 0.1$.

Method	"Relevant" & "Somewhat Relevant" as Gold Labels						"Relevant" as Gold Label					
	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR
Salience (max)	0.334	0.453	0.324	0.432	0.476	0.636	0.480	0.360	0.240	0.519	0.557	0.540
PPR	**0.390**$^{\blacktriangle}$	**0.613**$^{\blacktriangle}$	**0.347**	**0.476**$^{\triangle}$	**0.526**$^{\blacktriangle}$	**0.728**$^{\blacktriangle}$	**0.555**$^{\blacktriangle}$	**0.52**$^{\blacktriangle}$	**0.24**	**0.574**	**0.613**$^{\triangle}$	**0.633**$^{\blacktriangle}$
LTR	0.365	**0.667**	0.313	0.464	0.507	0.751	**0.561**	**0.606**	0.212	**0.57**	0.605	**0.669**
PPR	**0.433**$^{\triangle}$	**0.667**	**0.404**$^{\blacktriangle}$	**0.501**	**0.572**	**0.776**	0.541	0.545	**0.242**	0.561	**0.626**	0.668

Table 7: Performance by incremental plugging to LTR the most important features derived from Text Coherence and Salience groups. Statistically significant worsening between the system using all features (last row in the table) and the relative subset of them is indicated with $^{\blacktriangledown}$ for $p < 0.05$.

Feature	"Relevant" & "Somewhat Relevant" as Gold Labels						"Relevant" as Gold Label					
	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR	MAP	P@1	P@3	NDCG@5	NDCG@10	MRR
salience-max	0.351 $^{\blacktriangledown}$	0.631	0.374 $^{\blacktriangledown}$	0.497 $^{\blacktriangledown}$	0.549 $^{\blacktriangledown}$	0.748	0.382 $^{\blacktriangledown}$	0.523	0.256 $^{\blacktriangledown}$	0.536 $^{\blacktriangledown}$	0.578 $^{\blacktriangledown}$	0.650
+ *prior-stddev*	0.410 $^{\blacktriangledown}$	0.585	0.349 $^{\blacktriangledown}$	0.506 $^{\blacktriangledown}$	0.552 $^{\blacktriangledown}$	0.710 $^{\blacktriangledown}$	0.429 $^{\blacktriangledown}$	0.446	0.231 $^{\blacktriangledown}$	0.531 $^{\blacktriangledown}$	0.567 $^{\blacktriangledown}$	0.586 $^{\blacktriangledown}$
+ *prior-max*	0.471 $^{\blacktriangledown}$	0.738	0.410 $^{\blacktriangledown}$	0.571	0.610 $^{\blacktriangledown}$	0.814	0.454 $^{\blacktriangledown}$	0.554	0.267	0.590	0.619 $^{\blacktriangledown}$	0.673
+ *salience-cosine*	0.557	**0.754**	0.441	**0.645**	0.675	**0.836**	**0.632**	**0.615**	0.297	**0.693**	**0.708**	**0.733**
all features	**0.574**	0.708	**0.472**	0.629	**0.682**	0.815	0.609	0.569	**0.308**	0.654	0.696	0.710

importance score.[4] The model using only the four most important features belonging to Text Coherence and Salience achieves very similar performance to a more complicated solution, with no statistical significance difference among all metrics.

[4]The importance score is calculated as the number of splits a feature is used in the model that employs both Text Coherence and Salience groups of features.

9 CONCLUSION AND FUTURE WORK

In this work, we studied how to contextualize a trending entity by ranking related entities according to how properly they help to explain a trend. We proposed two main unsupervised and supervised solutions for addressing the problem at hand. These methods were experimented against different baselines on a novel dataset we built through a crowdsourcing methodology. For what concerns

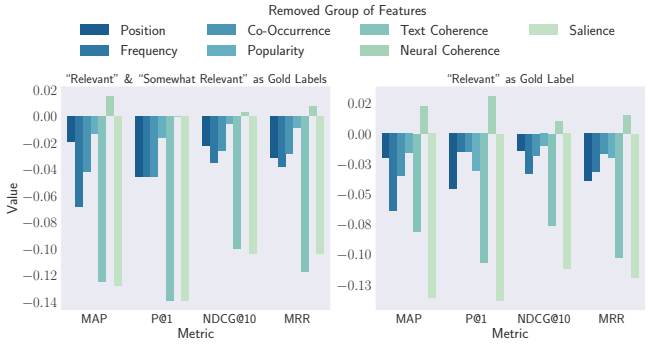

Figure 3: Feature ablation results where y-axes report the difference of performance between LTR using all features and without a specified group of features.

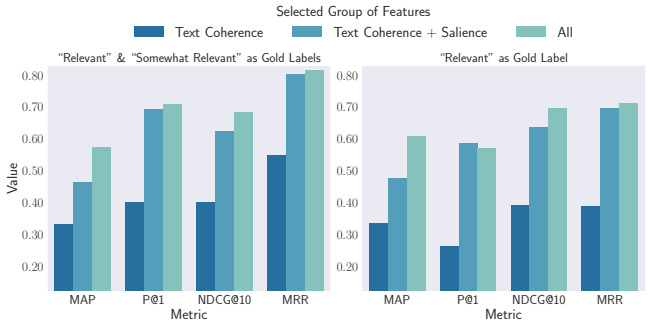

Figure 4: Performance of LTR solution using a subset of selected features.

future work, the quality of the ranking might be further improved by taking into account KB relationships, or by discovering salient OpenIE's facts [17] between the detected trending and contextual entities. Further development might also involve the application of our new research task in different downstream applications, such as entity-driven summarization, and query expansion.

REFERENCES

[1] Joshua Bambrick, Minjie Xu, Andy Almonte, Igor Malioutov, Guim Perarnau, Vittorio Selo, and Iat Chong Chan. 2020. NSTM: Real-Time Query-Driven News Overview Composition at Bloomberg. In *Proceedings of ACL*.

[2] Paolo Boldi, Massimo Santini, and Sebastiano Vigna. 2005. PageRank as a function of the damping factor. In *Proceedings of WWW*.

[3] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of SIGKDD*.

[4] Shruti Chhabra. 2014. Entity-centric summarization: generating text summaries for graph snippets. In *Proceedings of WWW*.

[5] Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of SIGIR*.

[6] Luciano Del Corro and Johannes Hoffart. 2020. Unsupervised Extraction of Market Moving Events with Neural Attention. In *CoRR*.

[7] Laura Dietz. 2019. ENT Rank: Retrieving Entities for Topical Information Needs through Entity-Neighbor-Text Relations. In *Proceedings of SIGIR*.

[8] Jesse Dunietz and Daniel Gillick. 2014. A new entity salience task with millions of training examples. In *Proceedings of EACL*.

[9] Paolo Ferragina and Ugo Scaiella. 2011. Fast and accurate annotation of short texts with wikipedia pages. *IEEE software* (2011).

[10] Darío Garigliotti, Faegheh Hasibi, and Krisztian Balog. 2017. Target type identification for entity-bearing queries. In *Proceedings of SIGIR*.

[11] Jheser Guzman and Barbara Poblete. 2013. On-line relevant anomaly detection in the Twitter stream: an efficient bursty keyword detection model. In *Proceedings of SIGKDD*.

[12] Taher H Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of WWW*.

[13] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in NIPS*.

[14] Daichi Koike, Yusuke Takahashi, Takehito Utsuro, Masaharu Yoshioka, and Noriko Kando. 2013. Time series topic modeling and bursty topic detection of correlated news and twitter. In *Proceedings of IJCNLP*.

[15] Craig Macdonald and Iadh Ounis. 2006. Voting for candidates: adapting data fusion techniques for an expert search task. In *Proceedings CIKM*.

[16] Nilamadhaba Mohapatra, Vasileios Iosifidis, Asif Ekbal, Stefan Dietze, and Pavlos Fafalios. 2018. Time-aware and corpus-specific entity relatedness. In *DL4KGS@ESWC*.

[17] Marco Ponza, Luciano Del Corro, and Gerhard Weikum. 2018. Facts that matter. In *Proceedings of EMNLP*.

[18] Marco Ponza, Paolo Ferragina, and Francesco Piccinno. 2019. Swat: A system for detecting salient Wikipedia entities in texts. *Computational Intelligence* (2019).

[19] Ridho Reinanda, Edgar Meij, Joshua Pantony, and Jonathan Dorando. 2018. Related Entity Finding on Highly-heterogeneous Knowledge Graphs. In *Proceedings of ASONAM*.

[20] Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia* (2008).

[21] Salvatore Trani, Claudio Lucchese, Raffaele Perego, David E Losada, Diego Ceccarelli, and Salvatore Orlando. 2018. SEL: A unified algorithm for salient entity linking. *Computational Intelligence* (2018).

[22] Nikos Voskarides, Edgar Meij, Ridho Reinanda, Abhinav Khaitan, Miles Osborne, Giorgio Stefanoni, Prabhanjan Kambadur, and Maarten de Rijke. 2018. Weakly-supervised contextualization of knowledge graph facts. In *Proceedings of SIGIR*.

[23] Ian H Witten and David N Milne. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. (2008).

[24] Chuan Wu, Evangelos Kanoulas, and Maarten de Rijke. 2019. It all starts with entities: A Salient entity topic model. *Natural Language Engineering* (2019).

[25] Chenyan Xiong and Jamie Callan. 2015. Query expansion with freebase. In *Proceedings of ICTIR*.

[26] Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards better text understanding and retrieval through kernel entity salience modeling. In *Proceedings of SIGIR*.

[27] Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji. 2018. Wikipedia2Vec: An Optimized Tool for Learning Embeddings of Words and Entities from Wikipedia. *CoRR* (2018).