Quality-Aware Entity-Level Semantic Representations for Short Texts

Wen Hua[#], Kai Zheng[#†], Xiaofang Zhou[#†]
[#]School of ITEE, The University of Queensland, Brisbane, Australia
[†]School of Computer Science and Technology, Soochow University, Suzhou, China
w.hua@uq.edu.au, {kevinz, zxf}@itee.uq.edu.au

Abstract

Recent prevalence of Web search engines, microblogging services as well as instant messaging tools give rise to a large amount of short texts including queries, tweets and instant messages. A better understanding of the semantics embedded in short texts is indispensable for various Web applications. We adopt the entity-level semantic representation which interpretes a short text as a sequence of mention-enity pairs. A typical strategy consists of two steps: entity extraction to locate entity mentions, and entity linking to identify their corresponding entities. However, it is never a trivial task to achieve high quality (i.e., complete and accurate) interpretations for short texts. First, short texts are noisy, containing massive abbreviations, nicknames and misspellings. As a result, traditional entity extraction methods cannot detect every potential entity mentions. Second, entities are ambiguous, calling for entity linking methods to determine the most appropriate entity within certain context. However, short texts are length-limited, making it infeasible to disambiguate entities based on context similarity or topical coherence in a single short text. Furthermore, the platforms where short texts are generated are usually personalized. Therefore, it is necessary to consider user interest and its dynamics overtime when linking entities in short texts. In this paper, we summarize our work on quality-aware semantic representations for short texts. We construct a comprehensive dictionary and extend traditional dictionary-based entity extraction method to improve recall of entity extraction. Meanwhile, we combine three novel features, namely content feature, social feature and temporal feature, to guarantee precision of entity linking. Empirical results on real-life datasets verify the effectiveness of our proposals.

1 Introduction

Recent decades have witnessed the flourishing of Web search engines, microblogging services, as well as instant messaging tools. This results in an increasing amount of short texts, i.e., length-limited poorly-structured natural language texts. Short texts embed invaluable knowledge. For example, companies can estimate public support for their products by analyzing query logs; governments can discover potential threat by monitoring tweet streams. In order to harvest knowledge from short texts, we need to go beyond raw texts and discover semantics. In this paper, we adopt entity-level semantic representation, namely recognizing entities

Bulletin of the IEEE Computer Society Technical Committee on Data Engineering

from short texts. More formally, given a short text s, we need to obtain a sequence of mention-entity pairs $\{\langle m_1, e_1\rangle, \langle m_2, e_2\rangle, ..., \langle m_l, e_l\rangle\}$ where m_i is an entity mention (i.e., a noun phrase) detected from s and e_i refers to a real-world entity recorded in large-scale machine-understandable knowledgebases. We use Wikipedia[1] as an example knowledgebase in the rest of the paper. Fig. 1 depicts an example of entity-level semantic representation wherein three mention-entity pairs are identified from the given tweet, namely $\{\langle m_1$ ="allen iverson", e_1 =*Allen Iverson (basketball)*$\rangle$, $\langle m_2$ ="michael jordan", e_2 =*Michael Jordan (basketball)*$\rangle$, $\langle m_3$ ="shaquille o neal", e_3 =*Shaquille O'Neal (basketball)*$\rangle\}$.

Figure 1: An example of entity-level semantic representation.

A typical strategy for obtaining entity-level semantic representation consists of two steps: entity extraction which locates entity mentions $\{m_1, m_2, ..., m_l\}$ in a given short text s, and entity linking which identifies the entity e_i that each mention m_i refers to. The quality of semantic representation determines the quality of knowledge discovered from short texts, which in turn affects user experience of the aforementioned Web applications. We consider two types of quality criteria in this work: *completeness* and *accuracy*. In other words, we aim to extract every possible entity mentions from a short text, and meanwhile find the most appropriate (i.e., semantically coherent with the context) entities for these mentions. However, the noisy, contextualized and personalized textual sources introduce some unique challenges for obtaining high quality entity-level semantic representations for short texts. In the following, we demonstrate quality issues in short texts with several examples, and discuss the limitations of existing methods to address these problems.

Data quality problem 1: noisy text. There have been extensive efforts on entity extraction which can be classified into two categories, namely linguistic-based and dictionary-based. Linguistic-based approaches incorporate linguistic features, such as capitalization, digitalization, punctuation, part-of-speech tags and so forth, into a machine learning model (e.g., Support Vector Machine [1, 2], Maximum Entropy Model [3, 4, 5], Hidden Markov Model [6] and Conditional Random Field [7]) to detect entity mentions. However, short texts are informal and do not always observe linguistic rules, which makes traditional linguistic features (e.g., capitalization) inapplicable to entity extraction from short texts. Dictionary-based approaches [10, 11] are becoming increasingly popular nowadays due to their simplicity and real-time nature. They extract entity mentions in a streaming manner by checking for existence or frequency of a noun phrase in a predefined dictionary of entity mentions. In particular, the widely-used Longest Cover method searches for longest noun phrases contained in the dictionary while scanning a text. Note that most dictionary-based approaches implicitly require noun phrases to exactly match at least one element in the dictionary. Whereas, short texts are noisy and full of abbreviations, nicknames, and misspellings. For example, "new york city" is usually abbreviated to "nyc" and known as "big apple". Hence, we need to add some flexibility to existing dictionary-based entity extraction methods to guarantee the completeness of semantic representations for short texts.

[1]Wikipedia data is publicly available at `https://dumps.wikimedia.org/enwiki/`

Data quality problem 2: entity ambiguity. A knowledgebase can be regarded as a huge collection of mentions and entities as well as mappings between them. Hence, we can obtain the entity that a mention refers to directly from such mapping information. However, there exists one-to-many mappings between mentions and entities. In other words, a specific entity mention can correspond to multiple real-world entities. For example. "jordan" could be a mention of *Jordan (country)*, *Air Jordan*, *Michael Jordan (basketball)*, as well as *Michael Jordan (machine learning expert)*. An accurate semantic representation requires to find the most appropriate entity for each mention detected in a short text. To this end, it is indispensable for entity linking methods to resolve entity ambiguity. Existing approaches to entity linking [12, 13, 14, 15, 16, 17, 18] are mostly content-based and targeted on documents. They utilize a combination of three features, namely entity popularity, context similarity and topical coherence, to estimate the weights of candidate entities. In particular, the *entity popularity* feature assumes that users' attention is usually focused on a small subset of entities, and hence the historical frequency information can be regarded as a hint for entity linking. Take the entity mention "jordan" as an example. It is more possible that users are talking about entity *Air Jordan* rather than *Jordan (country)* considering its relatively larger popularity. The *context similarity* feature measures mention-entity correspondence by calculating similarity between texts around entity mentions and documents describing entities in a knowledgebase. The *topical coherence* feature assumes that entities mentioned in a single document should be topically coherent, and handles mentions collectively by considering semantic relatedness between corresponding entities. Despite the satisfying performance these features have achieved in documents, the accuracy of entity linking decreases dramatically in short texts due to their length limitation [19]. Short texts cannot provide sufficient information to calculate context similarity accurately, nor can they provide enough mentions to derive a joint and interdependent entity assignment based on cross-entity relationships. Another factor that could affect the accuracy of entity linking in short texts is the personalized nature of Web applications where short texts are generated. Take microblogging services as an example. Users interested in basketball are more likely to talk about *Michael Jordan (basketball)*, while users interested in computer science tend to mention *Michael Jordan (machine learning expert)* in her postings. Consequently, it is necessary to take user interest into consideration when linking entities in such personalized short text dataset. [20, 21, 22] believe that users' interest is scattered in the messages they broadcast, and adopt a content-based method to discover user interest. They construct a graph model between candidate entities detected from the whole set of short texts published by a single user and conduct entity linking collectively. However, the topics of users' postings vary significantly, making it inaccurate to infer user interest from such a diverse stream of short texts. Furthermore, a large amount of users are information seekers with limited broadcasting history [23], which also increases the difficulty of learning their interest. Finally, users' interest can be influenced by recent events and change over time. For example, *Michael Jordan (basketball)* is more likely to be mentioned during NBA seasons while *Michael Jordan (machine learning expert)* is probably a better candidate entity when ICML (International Conference on Machine Learning) is being held. Therefore, the dynamics of user interest should also be considered, in order to guarantee the accuracy of semantic representations for personalized short text sources.

In this paper, we summarize our work on entity-level semantic representations for short texts, with a focus on how we resolve the aforementioned data quality problems and thus guarantee the quality (i.e., completeness and accuracy) of semantic representations. Fig. 2 illustrates an overview of our framework. We adopt the two-step strategy for obtaining semantic representations. Specifically, we construct a large-scale dictionary from existing knowledgebases and extend traditional dictionary-based methods to allow for approximate entity extraction from noisy short texts (Sec. 2); we combine three novel features, namely content feature, social feature and temporal feature, which are calculated based on some pre-collected statistical information to resolve entity ambiguity in short texts (Sec. 3). We empirically evaluate our framework on real-life datasets (i.e., queries and tweets) and present some of the experimental results in Sec. 4.

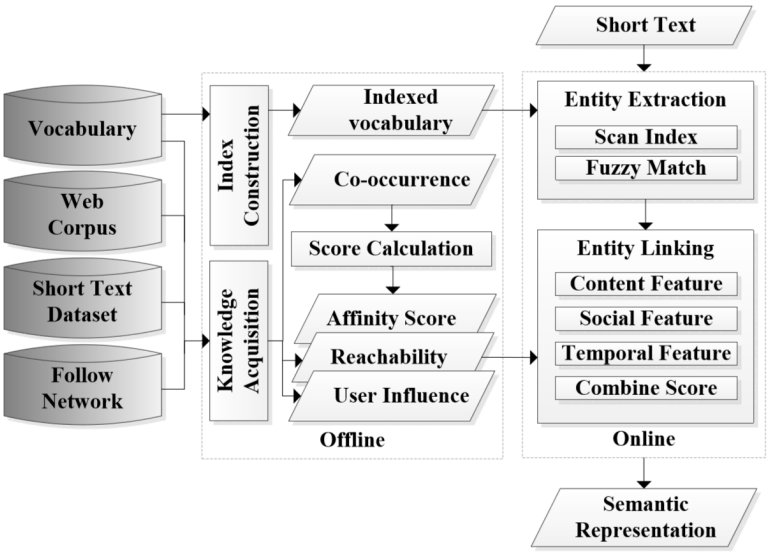

Figure 2: Framework overview.

2 Entity Extraction

Entity extraction is the first step for obtaining entity-level semantic representations. It detects entity mentions from texts written in a natural language. We adopt the widely-used dictionary-based approach to extract entity mentions, considering its simplicity and real-time nature. Since short texts are full of abbreviations, nicknames and misspellings, we need to handle this noise specifically, in order to achieve a complete set of entity mentions from a short text.

2.1 Handling abbreviations and nicknames

A large-scale dictionary (also called vocabulary, lexicon, or gazetteer) is a prerequisite for dictionary-based entity extraction, which is usually constructed from existing knowledgebases. Wikipedia is organized as a collection of Web pages including entity pages, disambiguation pages and redirect pages. Each entity page corresponds to a specific entity, and contains a detailed description about that entity. We can obtain the set of entities from Wikipedia' entity pages. Disambiguation page of a mention consists of a list of hyperlinks to entities it can refer to, from which we can extract the one-to-many mappings between mentions and entities. For example, given disambiguation page of "harry potter", we correspond mention "harry potter" to entities *Harry Potter (book)*, *Harry Potter (film)*, *Harry Potter (character)* and *Harry Potter (journalist)*. Redirect page is a virtual page which jumps to a specific entity page. Therefore, the URIs of redirect page and corresponding entity page can be used to extract abbreviations and nicknames of entities. For instance, the redirect page of "nyc" links to the entity page of "New York City", and hence "nyc" should be an abbreviation of entity *New York City*. Another information that can help reducing noise in short texts is the hyperlink structure between Wikipedia' entity pages. From the anchor texts of hyperlinks (e.g., "big apple") and the entity pages they point to (e.g., *New York City*), we can also obtain the abbreviations and nicknames of entities. In this way, we construct a dictionary which contains a huge collection of entity mentions along with their abbreviations and nicknames[2].

[2]The dictionary of abbreviations and nicknames is publicly available at `http://probase.msra.cn/dataset.aspx`

2.2 Handling misspellings

Approximate entity extraction is necessary to cope with misspellings in short texts. It locates substrings in a text that are similar to some dictionary entries. To quantify the similarity between two strings, many similarity functions have been proposed including token-based similarity functions (e.g., jaccard coefficient) and character-based similarity functions (e.g., edit distance). We choose edit distance as our similarity function since it is more suitable for handling misspellings.

We adopt and extend the trie-based method [30] for approximate entity extraction. That is, given an edit distance threshold τ, we divide each entity mention into $\tau + 1$ segments evenly. The pigeonhole principle guarantees that if a substring is similar to an entity mention with respect to τ, it must contain at least one segment of that mention. We build a segment-based inverted index on the entire dictionary, where the entries are segments and each segment is associated with an inverted list of entity mentions containing the segment. Given a short text, we adopt the search-extension algorithm proposed in [30] to find all possible mentions. In other words, we first enumerate every substring of a short text and check whether it matches a segment using the trie structure. In this way, we obtain a set of segments contained in the short text. Then for each segment and the corresponding substring, we extend the substring to a longer substring similar to a mention in the inverted list. The most notable limitation of the existing trie-based framework is that it utilizes one specific edit distance threshold τ. However, our dictionary contains a large amount of abbreviations as well as multi-word entity mentions which require different edit distance thresholds. For example, in order to recognize misspelled multi-word mentions, we sometimes need a large edit distance threshold of at least 2. But when we apply the same edit distance threshold to abbreviations, it will lead to mistakes (e.g., "nyc" and "ntu" will be regarded as similar). To this end, we extend the trie-based framework to allow for various edit distance thresholds at the same time. The problem is how to determine the value of τ for different entity mentions. It can be expected that τ depends on the length of mentions. In other words, the longer a mention is, the more possible it will be misspelled and the more mistakes there will be. Therefore, we collect a large-scale short text dataset from search engines and microblogging sites, and invite annotators to label misspelled mentions along with their edit distances. We observe a near step-like distribution between edit distance and mention length, which is then used as our guideline for determining edit distance threshold for different entity mentions.

3 Entity Linking

Entity linking resolves entity ambiguity, i.e., the one-to-many mappings between mentions and entities. In other words, it estimates the weights of candidate entities and finds the best entity for a given mention within certain context. As discussed in Sec. 1, traditional content-based entity linking approaches cannot be directly applied to short texts due to length limitation and personalized nature. We introduce three novel features to guarantee the accuracy of entities recognized from short texts. Formally, given the set of candidate entities $E_m = \{e_1, e_2, ..., e_n\}$ for a mention m published by user u, the weight of each entity $S(e)$ is a combination of content feature, social feature and temporal feature.

$$S(e) = \alpha \cdot S_{content}(e) + \beta \cdot S_{social}(u, e) + \gamma \cdot S_{temporal}(e). \quad (1)$$

In Eq. 1, α, β and γ ($\alpha + \beta + \gamma = 1$) are coefficients that represent relative contributions of content feature, social feature and temporal feature to the overall weighing function respectively, which can be manually defined or automatically learned using machine learning algorithms. We describe these three features in detail in the following sections.

3.1 Content feature

Short texts do not have sufficient content to calculate context similarity between mentions and candidate entities accurately. Meanwhile, the number of mentions that can be extracted from a short text are usually limited, making the topical coherence feature between entities inapplicable in short texts. As an alternative, we dig into semantic relatedness between any types of terms (e.g. verbs and adjectives, in addition to entities) to assist entity linking in short texts. Consider the tweet "wanna watch harry potter tonight" as an example. Only one mention "harry potter" can be detected, and hence we cannot apply topical coherence to determine the best entity for "harry potter" in this tweet. However, given the knowledge that the verb "watch" is much more semantically related to *Harry Potter (film)* than *Harry Potter (book)*, *Harry Potter (character)* and *Harry Potter (journalist)*, we can successfully identify *Harry Potter (film)* as the best entity for "harry potter" in this tweet according to such relatedness information.

The key technique here is to ensure the accuracy of relatedness calculation between terms. In this work, we consider relatedness in terms of both similarity and co-occurreence. That is, two terms are related if they are semantically similar or they frequently co-occur within certain context. Therefore, we propose an *affinity socre* $S_{affinity}(x, y)$ to denote semantic relatedness between two terms x and y, which is defined as the maximum value of similarity score and co-occurreence score.

$$\begin{aligned} S_{affinity}(x, y) &= \max(S_{sim}(x, y), S_{co}(x, y)) \\ &= \max(cosine(\vec{c}_x, \vec{c}_y), cosine(\vec{c}_{co(x)}, \vec{c}_y)). \end{aligned} \tag{2}$$

$S_{sim}(x, y)$ in Eq. 2 denotes semantic similarity between terms x and y, which is calculated by the cosine similarity between their category distributions $\vec{c}_x$ and $\vec{c}_y$, namely $S_{sim}(x, y) = cosine(\vec{c}_x, \vec{c}_y)$. Each entity is classified into several categories in Wikipedia. For example, entities *Michael Jordan (basketball)* and *Shaquille O'Neal (basketball)* are semantically similar, since they share a large amount of categories such as "NBA all-stars", "basketball players", "business people" and so on.

$S_{co}(x, y)$ in Eq. 2 represents co-occurreence score between terms x and y. Some existing knowledgebases, such as WordNet, have already incorporated information about co-occurrence or relatedness between terms. However, we observe that terms of different types co-occur with different context. For instance, the verb "watch" co-occurs with entity *Harry Potter (movie)*, while the entity *Watch* co-occurs with entity *Omega SA*. Therefore, a more accurate co-occurrence network should be constructed between terms with specific types. We observe that

- The more frequently two terms co-occur and the closer they locate in a certain sentence, the larger their semantic relatedness should be;
- Common terms (e.g., "item" and "object") are meaningless in modeling semantic relatedness, and thus should be penalized.

Based on these observations, we automatically analyze a large-scale Web corpus (e.g., Wikipedia entity pages or general Web pages) and compute co-occurrence strength based on such factors as frequency, distance, and tf-idf measure. In this way, we obtain a co-occurrence network between verbs, adjectives and entities[3]. During the construction of co-occurrence network, some co-occurring information might be missing due to limited coverage of the Web corpus. As demonstrated in Fig. 3, we cannot find a sentence in the Web corpus that contain both "watch" and "harry potter", and hence the co-occurring information between the verb "watch" and the entity *Harry Potter (film)* is missing. Such a phenomenon will affect the accuracy of semantic relatedness. Therefore, we transform the original entity-level co-occurrence network into a category-level co-occurrence network by mapping entities to their categories. The nodes in the category-level co-occurrence network are

[3]The co-occurrence network is publicly available at `http://probase.msra.cn/dataset.aspx`

verbs, adjectives and categories, and the edge weights are aggregated from the original network. Given the knowledge that "watch" co-occurs with category "film", we can indirectly recover the co-occurring relationship between "watch" and entity *Harry Potter (film)*. Let $\vec{c}_{co(x)}$ and $\vec{c}_y$ denote the set of categories x co-occurs with in the category-level co-occurrence network and the set of categories y belongs to respectively. We observe that the larger the overlapping between these two sets, the stronger the relatedness between terms x and y, namely $S_{co}(x, y) = cosine(\vec{c}_{co(x)}, \vec{c}_y)$.

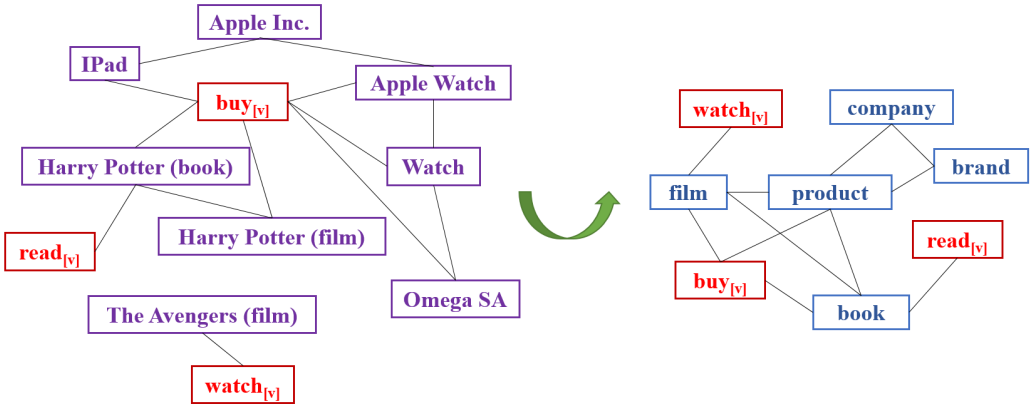

Figure 3: Examples of entity-level and category-level co-occurrence networks.

Based on Eq. 2, we can obtain semantical support, namely affinity score, of any contextual term recognized in short text s for candidate entity e. We choose the largest one as the content feature for entity linking.

$$S_{content}(e) = \max_{x \in s} S_{affinity}(x, e). \tag{3}$$

3.2 Social feature

In personalized web applications where short texts are generated, it is indispensable to consider user interest when conducting entity linking. As discussed in Sec. 1, traditional user interest modeling approaches based on historical broadcastings are inaccurate, due to the diverse range of topics embedded in messages and the existence of information seekers who tweet rarely. In this work, we resort to social interactions between users to indirectly infer user interest. We consider the "following" relationship in microblogging sites as an example, but our model can be easily extended to other platforms.

Microblogging users follow others to subscribe to tweets they are interested in. This means user u's interest in entity e can be reflected by her interest in following the set of users broadcasting about e. We define such a set of users as a *community* U_e which can be obtained by pre-processing a corpus of historical tweets using current state-of-the-art entity linking method [22]. We adopt reachability checking to estimate a user's interest in following another user, as formulated in Eq. 4. There are two issues we need to handle carefully to guarantee the accuracy of user interest estimation.

$$\begin{aligned} S_{social}(u, e) = S_{interest}(u, e) = S_{interest}(u, U_e) &= \frac{\sum_{v \in U_e} Reach(u, v)}{|U_e|} \\ &\approx S_{interest}(u, U_e^*) = \frac{\sum_{v \in U_e^*} Reach(u, v)}{|U_e^*|}. \end{aligned} \tag{4}$$

First, the *small-world* phenomenon in microblogging services [31] indicates that only reachable does not necessarily mean interested. Consequently, reachability which checks connectedness between two users should

be weighted, in order to achieve a more meaningful measurement of user interest. We consider both the distance and the strength of connection to weigh reachability between users. More formally,

$$Reach(u, v) = \frac{1}{d_{uv}} \cdot \frac{|F_{uv}|}{|F_u|} \tag{5}$$

In Eq. 5, d_{uv} is the shortest path distance from u to v. F_u denotes the collection of u's followees, and F_{uv} represents u's followees participating in at least one shortest path from u to v. Therefore, $\frac{|F_{uv}|}{|F_u|}$ actually reflects the strength of connection between u and v.

Second, different people have different influences in a community, and a user's interest in influential people contributes more to her interest in the community. To improve the accuracy of user interest estimation, we propose to detect a collection of most influential users for each community (denoted as U_e^*) and aggregate weighted reachability only with those influential users, as depicted in Eq. 4. Intuitively, a user is influential in a community associated with an entity e if:

- She is enthusiastic in broadcasting about entity e;
- She is discriminative among candidate entities E_m. This means an influential user should have a specific and continuous interest in broadcasting about entity e. For example, NBA's official account in Twitter (i.e., @NBAOfficial) hardly broadcasts about entities *Jordan (country)*, *Air Jordan* or *Michael Jordan (machine learning expert)*, making u's subscription to @NBAOfficial an important hint of her interest in basketball. Therefore, @NBAOfficial can be regarded as a discriminative and influential user in the community associate with entity *Michael Jordan (basketball)*.

Based on these heuristics, we propose a tfidf-based approach and an entropy-based approach to calculate user u' influence in the community associate with entity e. We consider the proportion of tweets published by user u about entity e to formulate the first heuristic in both approaches. As for the second heuristic, the tfidf-based approach measures the percentage of candidate entities u has mentioned in her tweets using the idf model, whereas the entropy-based approach examines the shape of probability distribution of u's historical tweets on candidate entities using the entropy model. In practice, it is common that an influential user in a community (say @NBAOfficial) occasionally tweets about candidate entities of other communities (say *Air Jordan*). Such an incident posting should not cause huge impact on her influence in the original community. In this sense, the entropy-based approach is superior to the tfidf-based approach in modeling user influence.

3.3 Temporal feature

As discussed in Sec. 1, users' interest is dynamic and can be influenced by recent events. Therefore, we need to capture the variation of user interest to further improve the precision of entity linking. Generally speaking, users are interested in entities involved in recent events or those attracting much public attention recently. We propose a temporal feature called *entity recency* to model an entity's recent popularity. Entity recency can be identified when a burst of tweets about that entity occurs during recent time period. In this work, we adopt a simple but effective approach to measure entity recency - sliding window. Formally, given a time window τ, we define entity recency using Eq. 6 where D_e^τ denotes the set of recently-published tweets about entity e.

$$S_{temporal}(e) = Recency(e) = \begin{cases} \frac{|D_e^\tau|}{\sum_{e_i \in E_m} |D_{e_i}^\tau|} & |D_e^\tau| \geq \theta_1 \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Besides a burst of tweets about entity e, recency can also be indirectly signified by that of related entities. For example, the recency of *Chicago Bulls* and *NBA* enhances that of *Michael Jordan (basketball)*. Similarly, increasing amount of tweets about *ICML* implies more attention on machine learning experts like Michael Jordan *(machine learning expert)*. Therefore, we propose a *recency propagation model* to incorporate mutual reinforcement of recency between related entities.

- Since entity recency is used as a feature for entity linking, it should not be propagated between candidate entities of the same entity mention, such as *Jordan (country)*, *Air Jordan*, *Michael Jordan (basketball)* and *Michael Jordan (machine learning expert)*
- If two entities are more topically related with each other, recency should be propagated between them in a larger extent;
- Only highly-related entities can reinforce each other's recency. This avoids extensive recency diffusion to slightly-related entities.

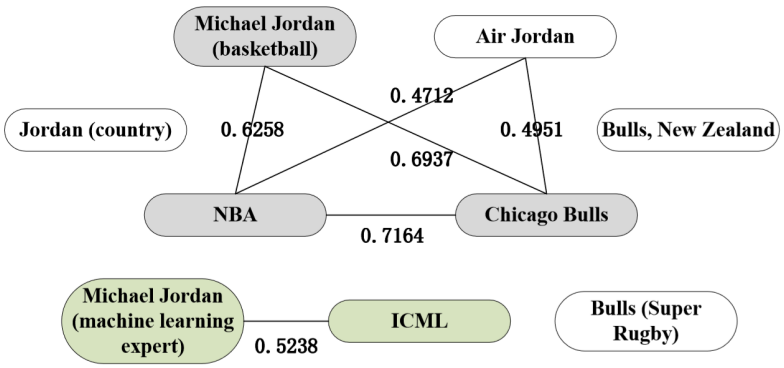

Figure 4: An example of recency propagation model.

Fig. 4 illustrates an example of the recency propagation model, which is formalized as an undirected graph on knowledgebase entities. Based on the above heuristics, edges are added only between highly-related entities corresponding to different entity mentions, and edge weights are defined based on semantic relatedness. We can use the affinity score described in Sec. 3.1 or the well-known Wikipedia Link-based Measure (WLM) [13] to model semantic relatedness between entities. Given the recency propagation model, we adopt a PageRank-like algorithm to combine recency gathered from underlying tweets and that reinforced by related entities.

4 Empirical Evaluation

We conducted extensive experiments on real-life datasets to evaluate the performance of our proposals. All the algorithms were implemented in C#, and all the experiments were conducted on a server with 2.90GHz Intel Xeon E5-2690 CPU and 192GB memory.

4.1 Benchmark

We briefly describe the dictionary and test datasets used in this work. In fact, our framework is generalized and can be applied to other dictionaries and short text platforms with slight extensions.

Dictionary. We downloaded the July 2014 version of English Wikipedia to build our dictionary for entity extraction. The Wikipedia dump contains 19.2 million entity pages, 6.3 million redirect pages, 0.2 million disambiguation pages, as well as 380 million hyperlinks between entity pages. Using the strategy described in Sec. 2.1 we obtained a huge dictionary of 29.3 million mentions including abbreviations and nicknames, and 19.2 million entities.

Test datasets. We constructed the test datasets by randomly sampling queries and tweets from a Web search engine (i.e., Bing) and a microblogging site (i.e., Twitter). We removed queries and tweets which contain entity mentions that cannot be recognized from our dictionary due to insufficient coverage. Altogether we obtained

1478 queries and 649 tweets. We also preprocessed the tweet dataset to remove some tweet-specific features such as @username, hashtags, urls, etc. We invited colleagues to annotate the test datasets, and the final labels were based on majority vote.

4.2 Effectiveness of proposals

Our empirical evaluation was divided into two parts according to the data quality problems discussed in Sec. 1. First, we evaluated whether our approach for entity extraction can effectively resolve textual noise, i.e., abbreviations, nicknames, and misspellings, in short texts. Second, we evaluated the performance of the proposed features, i.e., content feature, social feature, and temporal feature, compared with other features adopted in existing entity linking methods.

Effectiveness of entity extraction. Entity extraction locates entity mentions in a natural language text. In order to cope with abbreviations and nicknames in short texts, we construct a huge dictionary which incorporates not only entity mentions but also their abbreviations and nicknames using the strategy described in Sec. 2.1. We denote this dictionary as dic^*, and compare it with a preliminary dictionary dic containing only entity names. We use exact matching method to find entity mentions, and report the performance in terms of precision, recall and f1-measure in Table 1. Precision is the fraction of detected mentions that are labeled as correct, while recall is the fraction of correct mentions that are detected from the test dataset. More formally, precision $p = \frac{|M_{algo} \cap M_{label}|}{|M_{algo}|}$ and recall $r = \frac{|M_{algo} \cap M_{label}|}{|M_{label}|}$ where M_{algo} and M_{label} represent the set of entity mentions detected from the test dataset and those labeled by annotators respectively. F1-measure $f_1 = 2 \cdot \frac{p \cdot r}{p+r}$. From Table 1 we can see that exact matching based on dic^* can extract more entity mentions than dic, since dic^* enables matching method to recognize abbreviations and nicknames. And the increase of recall in the tweet dataset is slightly larger than in the query dataset, due to more frequent usage of abbreviations and nicknames in tweets. However, dic^* might cause more extraction errors sometimes. For example, given the information that "it" is an abbreviation of entity *Information Technology* in dic^*, exact matching will mistakenly recognize "it" in tweet "@payalpatel95 you're welcome! I never did it lol" as an entity mention. Such a phenomenon is especially common in the tweet dataset, since tweets are usually sentence-like while queries are keyword-like. This explains the much lower precision achieved in the tweet dataset than in the query dataset.

Table 1: Different dictionaries for entity extraction.

		precision	recall	f1-measure
query	dic	0.993	0.826	0.902
	dic^*	0.988	0.847	0.912
tweet	dic	0.707	0.682	0.694
	dic^*	0.627	0.718	0.669

We also apply approximate entity extraction to handle misspellings in short texts. We adopt and extend the trie-based method [30] to allow for approximate entity extraction with varying edit distance thresholds. We compare the performance of our approach (i.e., Trie with Varying edit distance, $TrieV$) with the trie-based method (i.e., $Trie$) and exact matching method (i.e., $Exact$), in terms of precision, recall, and f1-measure. From Table 2 we can see that approximate entity extraction can obtain more entity mentions from short texts than exact matching, at the cost of introducing slightly more extraction errors. By allowing for various edit distance thresholds depending on text length, $TrieV$ improves the precision of $Trie$ by reducing extraction errors caused by short entity mentions, abbreviations, etc. Overall, $TrieV$ achieves the highest f1-measure in both datasets. Note that the increase of recall in the tweet dataset is also larger than that in the query dataset, due to more misspellings in tweets.

Effectiveness of entity linking. We evaluate the performance of entity linking methods in terms of precision, and we only examine whether the correctly detected mentions are correctly linked. Formally, we calculate

Table 2: Different matching strategies for entity extraction.

		precision	recall	f1-measure
query	$Exact$	0.988	0.847	0.912
	$Trie$	0.943	0.922	0.932
	$TrieV$	0.984	0.918	0.950
tweet	$Exact$	0.627	0.718	0.669
	$Trie$	0.579	0.847	0.688
	$TrieV$	0.618	0.833	0.710

precision as $p = \frac{|M^*_{algo}|}{|M_{algo} \cap M_{label}|}$ where M^*_{algo} denotes the set of correctly linked mentions detected from the test dataset. Table 3 depicts the precision of entity linking based on five different combinations of features:

- [19]: consider topical coherence between entities.
- [22]: combine topical coherence between entities and user interest estimated from historical messages;
- $content$: consider semantic relatedness between any types of terms (e.g., verbs and adjectives, in addition to entities) based on co-occurrence relationship;
- $content + social$: combine semantic relatedness between any types of terms and user interest estimated through social interactions;
- $content + social + temporal$: combine semantic relatedness between any types of terms, user interest estimated through social interactions, as well as the dynamics of user interest overtime modeled as entity recency.

In Table 3, we only present the entity linking precision achieved by [19] and $content$ features for the query dataset, since we could not obtain author or timestamp information associated with each query when constructing this dataset which, however, is necessary to compute the social and temporal features. We obtain several observations from Table 3. First, the entity linking precision is consistently higher in the query dataset than in the tweet dataset. This is mainly because a large proportion of entities mentioned in tweets are celebrities and locations which are more ambiguous and harder to disambiguate. Second, $content$ performs better than [19] by considering topical coherence between any types of terms rather than only that between entities. Such a precision improvement is much more significant in the query dataset than in the tweet dataset, also due to the prevalence of celebrities and locations mentioned in tweets which cannot be disambiguated based on verbs or adjectives. Third, the precision of entity linking can be further increased by combining intra-tweet topical coherence with user interest information. Specifically, [22] discovers user interest from historical tweets and achieves larger precision than [19]. However, the improvement is limited due to the existence of information seekers in Twitter who cannot provide sufficient tweeting historical for interest estimation. Our proposed social feature, on the contrary, infers user interest based on social interactions, and hence increases precision to a larger extent. Fourth, the change of user interest overtime is also a crucial factor that should be considered when conducting entity linking in dynamic platforms such as microblogging sites. By combining all the three proposed features, namely content feature, social feature, and temporal feature, our framework achieves the best performance.

Table 3: Different features for entity linking.

	[19]	[22]	$content$	$content + social$	$content + social + temporal$
query	0.7104	-	0.8901	-	-
tweet	0.6667	0.6860	0.6777	0.7273	0.7315

5 Conclusion

Entity extraction and entity linking are basic steps for obtaining entity-level semantic representations for short texts. High quality, i.e., complete and accurate, semantic representations bring tremendous benefits to many Web applications. However, the noisy, personalized and dynamic nature of short text sources imposes unique challenges on both entity extraction and entity linking. In this paper, we summarize our work on semantic representations for short texts with a focus on the quality issues. Specifically, we construct a huge dictionary to incorporate abbreviations and nicknames, and extend the segment-based indexing structure on the dictionary to enable approximate entity extraction and thus reduce the impact of misspellings in short texts. Considering the prevalence of entity ambiguity in short texts and the limitations of traditional content-based entity linking approaches, we propose to combine three novel features, namely content feature (i.e., semantic relatedness between terms), social feature (i.e., user interest by social interactions), and temporal feature (i.e., entity recency which models the dynamics of user interest), to improve the accuracy of entity linking. Details on these features can be found in [24] and [25]. We report in this paper some of our empirical results on real-life datasets to examine the effectiveness of our framework in terms of precision and recall. The experimental results demonstrate significantly better performance of our proposals, compared with current state-of-the-art methods.

6 Acknowledgement

This work was partially supported by the ARC project under Grant No. DP140103171 and the NSFC project in Soochow under Grant No. 61472263.

References

[1] K. Takeuchi and N. Collier. Use of support vector machines in extended named entity recognition. In *CONLL*, pages 1–7, 2002.

[2] H. Isozaki and H. Kazawa. Efficient support vector classifiers for named entity recognition. In *COLING*, pages 1–7, 2002.

[3] H. L. Chieu and H. T. Ng. Named entity recognition: A maximum entropy approach using global information. In *COLING*, pages 1–7, 2002.

[4] O. Bender, F. J. Och, and H. Ney. Maximum entropy models for named entity recognition. In *CONLL*, pages 148–151, 2003.

[5] J. R. Curran and S. Clark. Language independent ner using a maximum entropy tagger. In *CONLL*, pages 164–167, 2003.

[6] G. Zhou and J. Su. Named entity recognition using an hmm-based chunk tagger. In *ACL*, pages 473–480, 2002.

[7] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CONLL*, pages 188–191, 2003.

[8] X. Liu, S. Zhang, F. Wei, and M. Zhou. Recognizing named entities in tweets. In *HLT*, pages 359–367, 2011.

[9] X. Liu, M. Zhou, F. Wei, Z. Fu, and X. Zhou. Joint inference of named entity recognition and normalization for tweets. In *ACL*, pages 526–535, 2012.

[10] C. Li, J. Weng, Q. He, Y. Yao, A. Datta, A. Sun, and B. Lee. Twiner: Named entity recognition in targeted twitter stream. In *SIGIR*, pages 721–730, 2012.

[11] D. M. de Oliveira, A. H. F. Laender, A. Veloso, and A. S. da Silva. Fs-ner: A lightweight filter-stream approach to named entity recognition on twitter data. In *WWW*, pages 597–604, 2013.

[12] R. Mihalcea and A. Csomai. Wikify!: Linking documents to encyclopedic knowledge. In *CIKM*, pages 233–242, 2007.

[13] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM*, pages 509–518, 2008.

[14] X. Han and J. Zhao. Named entity disambiguation by leveraging wikipedia semantic knowledge. In *CIKM*, pages 215–224, 2009.

[15] X. Han and J. Zhao. Structural semantic relatedness: A knowledge-based method to named entity disambiguation. In *ACL*, pages 50–59, 2010.

[16] S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *KDD*, pages 457–466, 2009.

[17] X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: A graph-based method. In *SIGIR*, pages 765–774, 2011.

[18] W. Shen, J. Wang, P. Luo, and M. Wang. Linden: Linking named entities with knowledge base via semantic knowledge. In *WWW*, pages 449–458, 2012.

[19] P. Ferragina and U. Scaiella. Tagme: On-the-fly annotation of short text fragments (by wikipedia entities). In *CIKM*, pages 1625–1628, 2010.

[20] A. Davis, A. Veloso, A. S. da Silva, W. Meira Jr., and A. H. F. Laender. Named entity disambiguation in streaming data. In *ACL*, pages 815–824, 2012.

[21] X. Liu, Y. Li, H. Wu, M. Zhou, F. Wei, and Y. Lu. Entity linking for tweets. In *ACL*, pages 1304–1311, 2013.

[22] W. Shen, J. Wang, P. Luo, and M. Wang. Linking named entities in tweets with knowledge base via user interest modeling. In *KDD*, pages 68–76, 2013.

[23] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *WebKDD/SNA-KDD*, pages 56–65, 2007.

[24] W. Hua, Z. Wang, H. Wang, K. Zheng, and X. Zhou. Short text understanding through lexical-semantic analysis. In *ICDE*, pages 495–506, 2015.

[25] W. Hua, K. Zheng, and X. Zhou. Microblog entity linking with social temporal context. In *SIGMOD*, pages 1761–1775, 2015.

[26] Y. Song, H. Wang, Z. Wang, H. Li, and W. Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, pages 2330–2336, 2011.

[27] D. Kim, H. Wang, and A. Oh. Context-dependent conceptualization. In *IJCAI*, pages 2654–2661, 2013.

[28] W. Wang, C. Xiao, X. Lin, and C. Zhang. Efficient approximate entity extraction with edit distance constraints. In *SIGMOD*, pages 759–770, 2009.

[29] G. Li, D. Deng, and J. Feng. Faerie: efficient filtering algorithms for approximate dictionary-based entity extraction. In *SIGMOD*, pages 529–540, 2011.

[30] D. Deng, G. Li, and J. Feng. An efficient trie-based method for approximate entity extraction with edit-distance constraints. In *ICDE*, pages 141–152, 2012.

[31] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.

[32] I. Witten and D. Milne. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *AAAI*, pages 25–30, 2008.