Socialized Language Model Smoothing via Bi-directional Influence Propagation on Social Networks

Rui Yan[1,2], Cheng-Te Li[3], Hsun-Ping Hsieh[4], Po Hu[1], Xiaohua Hu[1,5], and Tingting He[1]
[1]Computer School, Central China Normal University, Wuhan 430079, China
[2]Natural Language Processing Department, Baidu Inc., Beijing 100085, China
[3]Academia Sinica, Taipei 11529, Taiwan
[4]Air Force Institute of Technology, Kaohsiung 82047, Taiwan
[5]College of Computing and Informatics, Drexel University, Philadelphia, PA 19104, USA
{yanrui, phu, huxiaohua, tthe}@mail.ccnu.edu.cn
ctli@citi.sinica.edu.tw, sandoh714@gmail.com

ABSTRACT

In recent years, online social networks are among the most popular websites with high PV (Page View) all over the world, as they have renewed the way for information discovery and distribution. Millions of users have registered on these websites and hence generate formidable amount of user-generated contents every day. The social networks become "giants", likely eligible to carry on any research tasks. However, we have pointed out that these giants still suffer from their "Achilles Heel", i.e., extreme sparsity [34, 32]. Compared with the extremely large data over the whole collection, individual posting documents such as microblogs seem to be too sparse to make a difference under various research scenarios, while actually these postings are different. In this paper we propose to tackle the Achilles Heel of social networks by smoothing the language model via influence propagation. To further our previously proposed work to tackle the sparsity issue, we extend the socialized language model smoothing with bi-directional influence learned from propagation. Intuitively, it is insufficient not to distinguish the influence propagated between information source and target *without* directions. Hence, we formulate a bi-directional socialized factor graph model, which utilizes both the textual correlations between document pairs and the socialized augmentation networks behind the documents, such as user relationships and social interactions. These factors are modeled as attributes and dependencies among documents and their corresponding users, and then are distinguished on the direction level. We propose an effective learning algorithm to learn the proposed factor graph model with directions. Finally we propagate term counts to smooth documents based on the estimated influence. We run experiments on two instinctive datasets of *Twitter* and *Weibo*. The results validate the effectiveness of the proposed model. By incorporating direction information into the socialized language model smoothing, our approach obtains improvement over several alternative methods on both intrinsic and extrinsic evaluations measured in terms of perplexity, nDCG and MAP measurements.

WWW 2016, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4143-1/16/04.
http://dx.doi.org/10.1145/2872427.2874811.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing—*Text mining*; J.4 [**Social and Behavioral Sciences**]: Miscellaneous

General Terms

Algorithms, Experimentation, Performance

Keywords

Language model smoothing; bi-directional influence propagation; social networks

1. INTRODUCTION

Online social networking services allow users to have a convenient manner to interact with each other, especially on expressing themselves, sharing and distributing information through the underlying social networks [32, 34, 9]. People in *Twitter*[1] and *Facebook*[2] can write and comment succinct piece of postings to various kinds of memes (e.g., news, blog posts, images, and YouTube[3] videos) that spread fast from one to another. Likewise, geographical footprints (e.g., venues and attractions) can also be commented and shared with *Foursquare*[4] location-based service. We are surrounded by a social world with not only billion-scale [1] affiliated social peers but also the numerous proliferation of user-generated postings with short texts.

Although the social networking data has been utilized and validated to be effective in various applications, such as social search [26], recommender systems [31, 2], link prediction [8, 3], and information summarization [12, 35], none of existing studies aim at tackling the weakness of extreme sparsity problem, which had been pointed out in several literatures [32, 34]. The data sparsity problem results from the nature of social networking that a posted document is constrained to a fixed length (e.g. 140 characters in Twitter) for fast spreading. For such short postings, the data sparsity will make them rather indistinguishable. In other words, short postings lead to numerous unseen terms, which will be given zero probability values in the maximum likelihood estimator. Therefore, conventional language models [21, 10] may fail to represent short-text documents.

[1]https://www.twitter.com
[2]https://www.facebook.com
[3]https://www.youtube.com
[4]https://www.foursquare.com

Language Model Smoothing (LMS) is the most common approach to deal with the sparsity problem, and had been validated to be useful in various information retrieval tasks [21, 10]. The main idea for language model smoothing is to propagate term counts via a certain ways of projection to other places where they originally do not really exist [32, 34], which means to estimate potential term counts for documents without the terms by projecting from other documents. Several improved methods have been proposed, and can be roughly classified as 3 categories. The first is semantic association [29, 14, 25], which assumes documents with similar texts tend to share similar distribution of term counts. The second is positional proximity [37, 15, 29], which presumes that adjacent terms in sentences among documents can share similar count distributions. The third is social interactions [32, 34], which assumes the term usage behaviors of a user can be influenced by her friends or followees. That says, the term counts of a user have high potential to be close with those of her friends. However, the social-interaction strategy neglects that influence propagation can be direction-sensitive: the estimated influence should distinguish the *source* language model and the *target* language model. Between two users u and u', we cannot always assume that user u contribute the same influence for the propagation of term counts to user u' as that from u' to u. In other words, the influence of term usages between users and between documents should further incorporate the directional information, especially in online social networking services. In this paper, we aim to exploit bi-directional information of postings to provide socialized language model smoothing in a fine-granularity.

Hereby, we propose a socialized factor graph model to investigate various factors which could have impacts on language models, and measure the influence propagated along the factor graph with socialization. Given the influence estimated on the factor graph, we propagate the term occurrence in discounted counts and hence smooth the original language models. In this paper, the model is an extension of our previously proposed model in [34]. Unlike the socialized factor graph model without any direction information, we incorporate the bi-directional influence estimated along the graph. As we observe the influence from user u to u' is not equal to the influence from user u' to u, the intuition to characterize the bi-directional information since the propagated influence is generally asymmetric. We hence aim at distinguishing the influence propagated from u to u' in one way and the other way round from u' to u, rather than the integrated influence *between* users u and u' without directions as proposed in [34].

To the best of our knowledge, we are the first one to extend social influence with bi-directional information onto the textual dimension to facilitate the socialized language model smoothing. Our 1st contribution is to fuse a series of social attributes with textual information and form an integrated objective function, and moreover, we incorporate direction information into the model. Intuitively, the direction of the social ties should be used to distinguish source language model and target language model due to the asymmetry of the influence for propagation along with the social networks.

Another main technical challenge lies in how to define the attributes, factors and formulate the propagation functions to model the joint probabilistic factor graph. We explore several different factors captured from posting document pairs and user pairs, and evaluate their dependencies on each other. To be more specific, we have examined features such as text similarity, text quality, social status and social interactions and so on, and then grouped them together. Factor functions are finally formulated into one objective function to calculate the estimated influence and hence to smooth the language model accordingly.

In this paper, we tackle the problem of Bi-directional Socialized Language Model Smoothing (BSLMS) based on the factor graph model via bi-directional influence propagation. To evaluate the effectiveness of the new language model smoothing approach, we use two instinctively different social network datasets from *Twitter* and *Weibo*. Both of them are mainstream microblogs, English and Chinese. We apply intrinsic evaluation measured in perplexity and extrinsic evaluation in terms of nDCG and MAP. The experiments demonstrate our proposed bi-directional influence propagation based language model smoothing which distinguishes source and target language model could provide better modeling, and hence outperforms the direction-insensitive model. The result improvements indicate the effectiveness of our approach.

The rest of the paper is organized as follows. We start by introducing related work. Then we follow previous problem definition in socialized language model smoothing, and elaborate the bi-directional influence propagation based language model smoothing on factor graphs, using textual and social information in combination. We describe the experiments and evaluation in Section 5, and finally come to the conclusions.

2. RELATED WORK

Language models have been paid high attention to during recent years [21]. Many different ways of language modeling have been proposed to solve different research tasks. Better estimation of query language models [10, 11] and more accurate estimation of document language models [14, 25] have long been proved to be of great significance in information retrieval and text mining, etc. Language models are typically implemented based on traditional retrieval models, such as text weighting and normalization [36], but with more elegant mathematical and statistical foundations [22].

There is one problem for language models. Given limited data sampling, a language model estimation sometimes encounters with the zero count problem: the maximum likelihood estimator would give unseen terms a zero probability, which is not reliable. Language model smoothing is proposed to address this problem, and has been demonstrated to affect performance significantly [36, 10].

Many approaches have been proposed and tested. There are several ways of to smooth the original language model. The information of background corpus has been incorporated using linear combination [21, 36]. In contrast to the simple strategy which smoothes all documents with the same background, recently corpus contents have been exploited for more accurate smoothing. The basic idea is to smooth a document language model with the documents similar to the document under consideration through clustering [29, 14, 25]. Position information has also been used to enrich language model smoothing [37, 15] and the combination of both strategies of position and semantics [29]. In their work, the key idea is to define a language model for each position within a document, and score it based on the language models on all positions: hence the effect of positional adjacency is revealed. Beyond the semantic and/or position related smoothing intuitions, structural based language model smoothing is an alternative direction to investigate. A graph based language model smoothing method has been proposed utilizing structural adjacency only between neighboring nodes [17, 4].

There is a study in [13] which smoothes document language models of tweets for topic tracking in online text streams. Basically, it applies general smoothing strategies (e.g., Jelinek-Mercer, Dirichlet, Absolute Discounting, etc.) on the specific tracking task. Later, researchers have paid attention on language model smoothing with social information incorporated. Linear combination with social factor regularization [32] and factor graph model with social modeling [4, 34] are proposed to capture social influence for language model smoothing. To the best of our knowledge, we are the pilot study which characterizes more accurate language model smoothing via bi-directional social influence estimation, which literally distinguishes source and target language models. The bi-directional socialized language model smoothing is a novel insight.

3. PROBLEM FORMULATION

In this section, we introduce some of the preliminaries for language modeling and smoothing on social networks, and then we describe the problem of Bi-directional Socialized Language Model Smoothing (BSLMS) via factor graph model. The goal is to distinguish source and target language model and hence better to characterize the language smoothing schema on social networks.

Definition 1. (Document and Collection.) Given a posting document d_0 to smooth, we have a whole document collection $D = \{d_1, d_2, \ldots, d_n\}$ as the background set to smooth d_0.

In the context of web documents on social networks, e.g. Twitter or Weibo, etc., a particular user writes a posting document. Usually we smooth documents based on a corpus of plain texts, while for social networks, the posting documents are actually associated with more interesting elements. One of the most prominent elements is that users are connected through social ties such as *follower-followee* on microblog websites, or *friendship* on other websites. Through the social ties, the users can have more interactive actions towards the posting documents. For example, users can *comment*, *share* and *repost* documents from other users. In this way, there is a hidden network behind the document collection. User activities implicitly reflect more information behind the documents to model the textual properties. Social relationships are demonstrated to be useful to enhance textual information descriptions, and integrating the document contents and social information can disclose a more accurate estimation of the document language model to smooth [32, 34]. In this paper, we still keep microblog service as the basis for our study and hence make a good utilization of its characteristics for illustration. Specifically, given a posting document d_i, and its associated user u_i, together with the associated user networks, we give the following definition of socialized augmentation network.

Definition 2. (Socialized Augmentation Network.) For the posting documents and their corresponding users, we have a heterogeneous graph. We denote the whole graph G as a collection of nodes V and edges E, and have G=(V, E)=$(V_u, V_d, E_u, E_d, E_{u,d})$. It is obvious to see that there are three kinds of relationships associated: 1) (V_d, E_d) is a weighted directed graph between posting document pairs, where $V_d = \{d_i | d_i \in D\}$ is the posting collection with a size of $|D|$, and E_d is the set of relationships, indicating the influence from source postings to the target postings, which is our goal to estimate; 2) (V_u, E_u) is also a weighted directed graph indicating the social ties between users. $V_u = \{u_i | u_i \in V_u\}$ is the set of users with a size of $|V_u|$. E_u is established by the social behavior among users, which will be described in later sections; 3) $(V_{u,d}, E_{u,d})$ is the unweighted bipartite graph representing authorship of posting documents and users. $V_{u,d}$ = $V_u \cup V_d$. Edges in $E_{u,d}$ connect each posting document with all of its authors and help mapping from social dimension to textual dimension. Usually a posting document d is written by only one user u.

Definition 3. (Bi-Directional Social Influence.) In social networks such as Twitter or Weibo, the social ties between two users are generally not equal from the perspectives of both sides. For instance, the influence from the *follower* to the *followee* will be much greater than the influence from the other way round. Therefore, the influence on language models are asymmetric as well. To this end, we incorporate the concept of bi-directional social influence to distinguish source language models and target language models, rather than the conservative way to treat the social influence as identical for both sides of a particular social tie in [34]. Accordingly, the language model smoothing would have direction information as well.

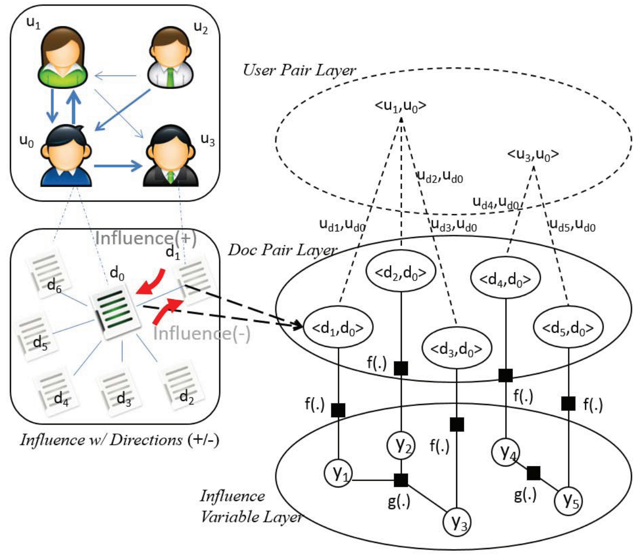

Figure 1: Graphical representation of the socialized augmentation factor graph with directions. The left part shows the heterogeneous graph, consisting of the document collection and users with social ties. The right part indicates the decomposable factor graph. The top layer shows the user pairs, which could be instantiated into several user pairs identified by different document pairs on the middle layer. The lower layer indicates the influence to estimate between document pairs. There are factor functions within the same layer ($g(.)$) and factor functions across layers ($f(.)$). In the final step, The language models are smoothed based on the influence estimated on the lower layer. Note that the influence is actually measured in bi-directions (illustrated as "+" and "-" in this figure), which indicates different social influence for different source-target language models.

We formally define BSLMS as follows:

Input: Given the entire document set D, and the socialization augmentation networks, we aim to smooth the language model of the target document, denoted as $P(w|d_0)$, based on the influence from all other documents d_i where $\{d_i | d_i \in D\}$. Since the influence is bi-directional, we use influence(+) to indicate the influence from d_i to d_0, and influence(-) as influence from d_0 to d_i. Hence the actually influence propagated from d_i to d_0 would be calculated by the gap between influence(+) and influence(-).

Output: The smoothed language model of $P(w|d_0^\star)$ for every original document d_0.

With these preliminaries, we show that relationships from document pairs, user pairs, user-document pairs and direction information can be all formulated into features and functions on the factor graph with a combined objective function.

4. METHODOLOGY

Here we propose a socialized factor graph to compute bi-directional influence propagation, and formulate the socialized language model smoothing problem into a unified learning framework. The model simultaneously incorporates all resources (i.e., texts and social information) into the augmentation contexts to generate high-quality estimation for document language models after smoothing.

The BSLMS problem contains several sub-problems: 1) directional influence measurement on document pairs, 2) bi-directional influence measurement on user pairs, and 3) bi-directional influence measurement among variables. We aim to quantify the correlation between document pairs based on semantic association derived from contents, while also we intend to augment the pairwise relationship between documents from the interactions of users on social level. We also analyze the dependency of variables on each other based on the same authorship on the variable layer. Furthermore, we characterize the direction of the influence propagation and hence we can better utilize social contexts with rich information on the documents to smooth the original language model. We apply to a series of related research tasks. The framework is illustrated in Figure 1, and the details are explained later.

4.1 Proposed Model

Factor graph assumes observation data are cohesive on both features and relationships between each other [7]. It has been successfully applied in many applications, such as social influence analysis [24, 23], social relationship mininig [3, 28], and linked data disambiguation [8, 27]. In this work, we formulate the social features and associated networks into the factor graph model, which is shown in Figure 1. Given the document pairs, let E_d be the pairwise links between two posting documents, and E_u be the user social ties. The input of the factor model is the whole document collection and the socialized augmentation contexts, and a target document d_0 to smooth. Both pairs are digested into the attribute factors, which are observable. There is also a set of hidden variables $Y = \{y_i\}_{i=1}^n$, representing the influence inferred from the observed pairs and coordination among the hidden variables. As for each $y_i \in Y$, there is a vector of y_i=$\{y_i^+, y_i^-\}$ indicates the influences on two directions. Each element represents the influence between the document pair with respect to influence(+) and influence(-).

We define two feature functions in the proposed factor model: *attribute factor* and *dependency factor*.

• **Attribute Factor**. The influence from a posting document to another could be estimated by some attributes (represented as **x**), which refer to features that are inherent to the documents and their authors. In general, we define a series of features for the document pairs and user pairs. These features include the textual based contents such as text quality, similarity and popularity, as well as the social ties such as user relationships, interactions, authoritativeness and so on. Details of the defined features are given in the next section. We use the feature function $f(y_i^+, \mathbf{x}_i)$ and $f(y_i^-, \mathbf{x}_i)$ to represent the posterior probability of label y_i given $\mathbf{x}_i$ contained in the pairwise information among the heterogenous nodes. "+" and "-" denote directions.

We define this type of potential function as a linear exponential function and to estimate the significance of each feature, we introduce a vector of weight variable α for each feature c, and formally we could define the attribute factors as the local entropy as follows:

$$
\begin{aligned}
f_i(y_i^+, \mathbf{x}_i) &= \frac{1}{Z_\alpha} \exp\{\sum_c \alpha_c f_{i,c}(y_i^+, x_{i,c})\} \\
f_i(y_i^-, \mathbf{x}_i) &= \frac{1}{Z_\alpha} \exp\{\sum_c \alpha_c f_{i,c}(y_i^-, x_{i,c})\}
\end{aligned} \tag{1}
$$

where $x_{i,c}$ is the c-th attribute to calculate the influence. $f_c(.)$ is the function to calculate the result from the c-th feature and α_c is the corresponding weight. Z_α is a normalization factor.

• **Dependency Factor**. As proposed, we introduce factors that are capable of handling multiple hidden variables on the variable layer, to characterize the dependencies among the posting documents generated by the same user. The dependency factor is to propagate the social influence among all posting documents from the same user. The heuristics is that if document d_0 is influenced by document d_i. It is highly possible to be influenced by the document d_k from the same user, which is actually a belief propagation.

To capture this intuition, we define the potential function to model the correlation of a candidate variable y_i with another candidate variable y_k in the factor graph. The function is defined as:

$$
\begin{aligned}
g(y_i^+, y_k^+) &= \frac{1}{Z_{ik,\beta}} \exp\{\beta_k g_k(y_i^+, y_k^+)\} \\
g(y_i^-, y_k^-) &= \frac{1}{Z_{ik,\beta}} \exp\{\beta_k g_k(y_i^-, y_k^-)\}
\end{aligned} \tag{2}
$$

where $g(.)$ is a function indicating whether two variables are correlated or not. Note that if $g(.)$=0, there will be no dependency between the two variables. In other words, the two variables are not correlated. Actually we can group the document set into clusters and each cluster is associated with one user, and we use Y_i to denote the cluster where y_i is in. Hence, for $\forall y_k \in Y_i$, y_i has dependency on y_k. We further revise Equation (2) as:

$$
\begin{aligned}
g_i(y_i^+, Y_i) &= \prod_{y_k \in Y_i} g(y_i, y_k) = \frac{1}{Z_\beta} \exp\{\sum_{y_k \in Y_i} \beta_k g_k(y_i^+, y_k^+)\} \\
g_i(y_i^-, Y_i) &= \prod_{y_k \in Y_i} g(y_i, y_k) = \frac{1}{Z_\beta} \exp\{\sum_{y_k \in Y_i} \beta_k g_k(y_i^-, y_k^-)\}
\end{aligned} \tag{3}
$$

Again, Z_β is the normalization factor.

In this way, the influence estimation could be viewed as a sequence labeling process [35], i.e., the judgment on a certain pair is affected by the "similar" pairs (i.e., the documents written by the same user in this work), which is exactly the intuition for language model smoothing [17].

• **Objective Function**. In general, the attribute factors capture the potential influence from document/user pairs and the dependency factor captures correlations between variables. In Equation (1) we define the features $f_c(.)$ for all attributes, where α_c is the corresponding weight. In Equation (3), we define the correlation where β_k indicates the weights. Let Y and X be the sets of candidate variables and attribute variables respectively, we define a joint probability encoded within the factor graph model by multiplying all potential functions and can be written as

$$
\begin{aligned}
P_\theta(X, Y) = &\frac{1}{Z} \prod_{i=1}^{N} f_i(y_i^+, \mathbf{x}_i) f_i(y_i^-, \mathbf{x}_i) \\
&\times \prod_{i=1}^{N} g_i(y_i^+, Y_i) g_i(y_i^-, Y_i)
\end{aligned} \tag{4}
$$

By integrating the defined factor functions, and also following the labeling assumption [6], we can define the following likelihood over all the undetermined labels of all instances, i.e., Y=$\{y_i\}_{i=1}^n$. The objective function sums up log-likelihood:

$$
\begin{aligned}
&\mathcal{O}(\theta) = \log P_\theta(X, Y) \\
&= \sum_i \sum_c \alpha_c \Big(f_{i,c}(y_i^+, x_{i,c}) + f_{i,c}(y_i^-, x_{i,c}) \Big) \\
&+ \sum_{y_k \in Y_i} \beta_k \Big(g_k(y_i^+, y_k^+) + g_k(y_i^-, y_k^-) \Big) - \log Z
\end{aligned} \tag{5}
$$

$Z = Z_\alpha Z_\beta$ is the normalization factor, which sums up the likelihood of P_θ over all instances. θ is the collection of parameters indicating weights, i.e., $\theta = \{\alpha\} \cup \{\beta\}$. $f(.)$ denotes the factor function and $g(.)$ indicates the dependency factor. Calculating the probability for each factor (in deriving the log-gradient of the objective function) requires a loopy sum-product inference algorithm. With the learned parameters, we may estimate an undetermined influence between document pairs in the test set by inferring the bi-directional influence propagated and then smooth language models accordingly. The inference algorithm is introduced in section 4.3.

4.2 Function Definitions

Many features have been designed for document correlation and social associations among users in previous literature. In this paper, we investigate 9 features or factors. Note that these features investigated, textual features and social features, are used in previous works and they are not that novel. To be self-contained, we still introduce them from one to another. We start from the feature definition first.

4.2.1 Attributes

We use the potential functions in the factor graph model to learn the potential influence for a document d_i to cast on d_0. Referring to Equation (1), we define the attribute functions as follows:

Text Similarity. It is intuitive that the textual similarity between two documents play an important role in language model smoothing [29, 14, 25]. Similar documents should be smoothed with higher weights since it is more consistent with their existing models. We use the cosine similarity between two unigram models:

$$f_{sim} = \frac{d_0 \cdot d_i}{||d_0||||d_i||} \tag{6}$$

Text Quality. We also measure the text quality of document d_i. It is not a good idea to smooth the target language model using a piece of text with low quality. Hereby we use the Out-Of-Vocabulary (OOV) ratio to measure the textual quality. The lower OOV ratio, the higher quality would be. Against the vocabulary from the official news corpora [38], OOV in microblogs often refers to a set of misspellings, informal terminologies, and irregular symbols.

$$f_{oov} = 1 - \frac{|\{w|w \in (d_i \cap \text{OOV})\}|}{|d_i|} \tag{7}$$

Technically, the measurement of *text quality* is not a pairwise function strictly between d_0 and d_i, but the criteria is indeed a practical indicator to decide whether or not to propagate the influence from d_i to d_0. We also include similar criteria for user pairs.

Posting Popularity. It is intuitive that a popular posting document is more likely to influence on many other posting documents. We use the aggregated numbers of social interaction (i.e., replies and retweets) as the approximation of popularity for d_i.

Social Status. Since micro-blogging service requires no reciprocal linkage among users, it is natural to assume that the social status is asymmetric between two users. A followee is more likely to influence the followers. This feature is represented by nominal values, e.g., '1' - the user of d_0 follows the user who writes d_i; '-1' - the user of d_0 is followed by the user who writes d_i; '0' - the two users have no direct follow behaviors.

User Similarity. We presume that people within the same social circle will have a larger probability to influence each other. Still, due to the asymmetry, we measure the Jaccard distance of the common followees of two users as their similarity. We use function $\mathcal{F}(u)$ to denote the social circle set for the user u. The $\mathcal{F}(.)$ of "followee" can be replaced by "followers" or extended to "friends".

$$f_{usim} = \frac{|\mathcal{F}(author(d_0)) \cap \mathcal{F}(author(d_i))|}{|\mathcal{F}(author(d_0)) \cup \mathcal{F}(author(d_i))|} \tag{8}$$

Interaction Strength. We also include the strength of interactions between the user pairs. It is possible that if two users have frequent social interactions, they are likely to share the writing preference on the term usage. Due to the asymmetrical social relationship, we only count the times for $author(d_0)$ to repost and comment from $author(d_i)$ to measure how likely for the user to be influenced.

Repost Behavior. Due to the special phenomenon of *repostings* on Twitter and Weibo, we have some initial indicators to distinguish the bi-directional influence. For each retweeted pairs, the target language model have almost the same contents and hence we label the influence from the reposted document as 1 and the influence as 0 in the other way. To be precise, if d_0 is a reposting from d_i, we measure $y_i^+ = 1$ and $y_i^- = 0$.

User Impacts. On social networks, some users are intrinsically have much larger influence on the others, e.g., sports stars, political celebrities, etc. Their words are usually copied, quoted and spreaded. There are many different ways to evaluate the user impacts, while we use the classic PageRank [19] scores to denote user impacts. The linkage based PageRank algorithm is quite suited to the scenario of user impact measurement. With a large number of in-links, the user is almost guaranteed to have high social impacts.

4.2.2 Dependency

As for the dependency function between candidate variables, referring to Equation (3), we define the function $g(.)$ for two candidate variables associated by the same user authorship in Y_i, and let the corresponding variables be y_i and y_k, respectively. The dependency function aims at encoding the influence propagation between posting documents from the same user, defined as follows.

$$\begin{aligned} g_k(y_i^+, y_k^+) &= \frac{y_k^+}{\sum_{j \in NB(i)} y_j^+} \\ g_k(y_i^-, y_k^-) &= \frac{y_k^-}{\sum_{j \in NB(i)} y_j^-} \end{aligned} \tag{9}$$

where $NB(i)$ represents all of the neighboring nodes of node i. The above definition of the dependency function is a normalization from both directions of the influence so that we can propagate the influence among the network. Although it is feasible to include all the neighbours around a certain node to measure the dependence in theory, we still follow the intuition that the documents from the same author should be more likely to carry such dependency of influence propagation [34]. Hereby we only measure the dependency correlation when $author(d_i)$=$author(d_k)$.

4.3 Model Inference

To train the model, we can take Equation (5) as the objective function to find the parameter configuration that maximizes the objective function. While it is intractable to find the exact solution, approximate inference algorithms such as sum-product algorithm [7, 24], can be used to infer the variables $\mathbf{y}$.

In sum-product algorithm, messages are passed between nodes and functions. Message passing is initiated at the leaves. Each node v_i remains idle until messages have arrived on all but one of the edges incident on the node v_i. Once these messages have arrived, node v_i is able to compute a message to be sent onto the one remaining edge to its neighbor. After sending out a message, node v_i returns to the idle state, waiting for a "return message" to arrive from the edge. Once this message has arrived, the node is able to compute and send messages to each of neighborhood nodes. This process runs iteratively until convergence [24].

However, traditional sum-product algorithm cannot be directly applied for bi-directional propagation. We hereby consider a basic extension of the sum-product algorithm: directional sum-product. The algorithm iteratively updates a vector of messages m between variable nodes and factor (i.e. feature function) nodes. Hence, two update rules can be defined respectively for the direction-aware ('+' and '-') message sent from variable node to factor node and for the message sent from factor node to variable node.

The *Product* part:

$$m_{y\to f}(y^+) = \prod_{f'\sim y\backslash f} m_{f'\to y}(y^+) \prod_{f'\sim y\backslash f} m_{f'\to y}(y^-)^{(\tau(+,-))}$$
$$m_{y\to f}(y^-) = \prod_{f'\sim y\backslash f} m_{f'\to y}(y^+)^{(\tau(+,-))} \prod_{f'\sim y\backslash f} m_{f'\to y}(y^-) \tag{10}$$

The *Sum* part:

$$m_{f\to y}(y^+) = \sum_{\sim\{y\}} \left(f(Y^+) \prod_{y'\sim f\backslash y} m_{y'\to f}(y'^+) \right) + \tau(+,-) \sum_{\sim\{y\}} \left(f(Y^+) \prod_{y'\sim f\backslash y} m_{y'\to f}(y'^+) \right) \tag{11}$$

and

$$m_{f\to y}(y^-) = \tau(+,-) \sum_{\sim\{y\}} \left(f(Y^-) \prod_{y'\sim f\backslash y} m_{y'\to f}(y'^{,-}) \right) + \sum_{\sim\{y\}} \left(f(Y^-) \prod_{y'\sim f\backslash y} m_{y'\to f}(y'^{,-}) \right) \tag{12}$$

where

• $f' \sim y\backslash f$ represents f' is a neighbor node of variable y on the factor graph except factor f;

• Y is a subset of hidden variables that feature function f is defined on; for example, a feature $f(yi; yj)$ is defined on edge e_{ij}, then we have $Y = \{y_i, y_j\}$; $\sim \{y\}$ represents all variables in Y except y;

• the sum $\sum_{\sim\{y\}}$ actually corresponds to a marginal function for y on one of the direction (+ or -) of the influence;

Note that the coefficient $\tau(.)$ represents the correlation between the two directions (+ and -). This function can be defined in a variety of ways. Intuitively, the influence on such two directions could have some dependencies due to some of the features and characteristics in common. In this work, we assume both ways of influences are independent from each other for simplicity, which means $\tau(+,+)$ = 1 and $\tau(-,-)$ = 1, while $\tau(+,-)$ = 0.

Finally, the bi-directional socialized language model smoothing is estimated on the bi-direction level influence propagation.

$$y = \begin{cases} y^+ - y^- & y^+ \geq y^- \\ 0 & y^+ < y^- \end{cases} \tag{13}$$

In this way, we obtain the estimated influence value with respect to each direction calculated in the last iteration and then apply into language model smoothing.

4.4 Language Model Smoothing

Given the estimated influence from all other posting documents, we now propose a term-level language model smoothing approach based on bi-directional influence propagation. Each word propagates the evidence of its occurrence to other documents based on the estimated influence. To capture the *proximity* heuristics used in language model smoothing [17, 15, 25, 29], we assign "close-by" words with higher propagated counts than those ones which are "far away" from each other. In other words, most propagated counts come from "nearby" terms, while higher influence indicates closer proximity [33, 30, 39].

In general, a specific posting document can be smoothed using the background information from the document collection. The traditional way is to concatenate the document language model $P(w)$ and the background $P_B(w)$ using linear combination, i.e., $P'(w)$=$(1 - \lambda)P(w)$+$\lambda P_B(w)$ where λ is the damping factor. Hereby we propose to smooth the language model based on the term level in a finer-granularity.

The idea for term projection is that if a word w occurs at a particular posting document, we would like to assume that the highly influenced document (i.e., with influence +) will also have the words occurred, with a discounted count. The larger the influence estimated, the larger the propagated term counts there will be. Note that each propagated count has a value less than 1.

Let $d_0 = \{w_1, w_2, \ldots, w_{|d_0|}\}$ where $|d_0|$ is the length of the document. We use $c(w, d_0)$ to denote the original term count within document d before smoothing. If w does not occur in d_0, $c(w, d_0)$ will be 0, which is a zero count problem. We have calculated the values of Y=$\{y_i\}_{i=1}^n$ from Equation (13) in the last section, indicating the influence from the d_i to the document d_0 to smooth. The function actually serves as a discounting factor for terms measured in pairwise. We use $c'(w, d_0)$ to denote the total propagated count of term w from its occurrences in all other documents, i.e.,

$$c'(w, d_0) = (1 - \lambda)c(w, d_0) + \lambda \sum_{y_i \in Y} y_i \cdot c(w, d_i) \tag{14}$$

Even if $c(w, d_0)$ is 0, $c'(w, d_0)$ may be greater than 0.

Based on term propagation, we have a term frequency vector $\{c'(w_1, d_0), \ldots, c'(w_v, d_0)\}$ for the virtual document $d_0^\star$ extended from document d_0. We store the term information with calculated influence in this vector. Thus the language model of this new **smoothed** virtual document can be estimated as

$$P(w|d_0^\star) = \frac{c'(w, d_0^\star)}{\sum_{w'\in V} c'(w', d_0^\star)} \tag{15}$$

V is the vocabulary set and $\sum_{w'\in V} c'(w', d_0^\star)$ is the length of the virtual document after smoothing.

5. EXPERIMENTS AND EVALUATION

5.1 Datasets and Experimental Setups

We use the data crawled from the online social networks through the "following" linkage established in [34, 32]. The datasets are consisted of microblogs and the corresponding users, which form the heterogeneous network. The datasets monitored Twitter data from 3/25/2011 to 5/30/2011, and Weibo data from 9/29/2012 to 11/30/2012. We use roughly one month as the training set and the rest as testing set. The details of the data are listed in Table 1.

Pre-processing. Basically, the social network factor graph can be established from all posting documents and all users, however, the data is noisy. We first pre-filter the pointless babbles [1] by applying the the linguistic quality judgements (e.g., OOV ratio) [20],

Table 1: **Statistics of the social network datasets.**

	#User	#Document.	#Link	Language
Twitter	9,449,542	364,287,744	596,777,491	English
Weibo	3,923,021	216,302,309	258,543,931	Chinese

and then remove inactive users that have less than one follower or followee and remove the users without any linkage to the remaining posting documents. We remove stopwords and URLs, perform stemming and segmentation (for Chinese texts), and build the graph after filtering, and estimate variable values on the factor graphs. We establish the language model smoothed by the estimated influence.

5.2 Algorithms for Comparison

To illustrate the performance, we implement several alternative algorithms as baselines to compare with our method. The baselines include naive smoothing, smoothing by semantics, positional smoothing and socialized language model smoothing from very recent studies. For fairness we conduct the same pre-processing procedures for all algorithms.

The first baseline is based on the traditional language model: **LM** is the language model without smoothing at all. We include the plain smoothing of **Additive** (also known as Add-δ) smoothing and **Absolute Discouting** which decreases the probability of seen words by subtracting a constant [18]. We also implement several classic strategies smoothed from the whole collection as background information: **Jelinek-Mercer** applies a linear interpolation, and **Dirichlet** employs a prior on collection influence [36, 10].

Beyond these simple heuristics, we examine a series of semantic based language model smoothing. The most representative two semantic smoothing methods are the Cluster-Based Document Model (**CBDM**) proposed in [14], and the Document Expansion Language Model (**DELM**) in [25]. Both methods use semantically similar documents as a smoothing corpus for a particular document: CBDM clusters documents beforehand and smooths a document with the cluster where it belongs to, while DELM finds nearest neighbors dynamically for the document as the smoothing cluster. However, both methods are only based on document-level semantic similarity. We also include Positional Language Model (**PLM**) [15], which is the state-of-art positional proximity based language smoothing. PLM mainly utilizes positional information without semantic information. We implemented the best reported PLM configuration.

For the last baseline group, we include the state-of-art socialized language model smoothing methods, i.e., Social Regularized Smoothing (**SRS**) [32], Cold-Start Personalized Language Model (**CSPLM**) [4] and Socialized Language Model Smoothing (**SLMS**) [34]. All these approaches managed to utilize social information for smoothing from different aspects, but neither distinguishes direction information. We compare our proposed bi-directional socialized language model smoothing (**BSLMS**) against all these baselines to verify the effect of the proposed model with the directional information in addition.

5.3 Evaluation Metric

It is generally difficult to examine the effect of language model directly [29, 25, 15]. For most of the language model smoothing research, the performance is measured based on extrinsic evaluations (e.g., retrieval) [29, 15, 32, 34]. We include an extrinsic evaluations in this study, i.e., standard posting document retrieval, but first we aim to evaluate the information contained in the language

Hashtag Clusters	Numbers	Notes
1. apple	42,528	Tech: apple products
2. nfl	40,340	Sport: American football
3. travel	38,345	General interst
4. mlb	38,261	Sport: baseball
5. fashion	30,053	General interest
1. 中国好声音	72,184	TV show: voice of China
2. 舌尖上的中国	71,169	Food: Chinese foods
3. 微博	63,154	Tech: Microblog service
4. 爱情公寓	57,783	TV drama: culture
5. 小米	49,428	Tech: smart phone

Table 2: Clusters of hashtag topics explored in our study.

itself. Hence we use language *perplexity* to evaluate the smoothed language model.

5.3.1 Intrinsic Evaluation

Our first set of experiments involved intrinsic evaluation of the "perplexity" approach based on a clustering scenario. The experimental procedure is as follows: we manually selected 10 topics (5 for each dataset) based on popularity (measured in the number of postings) and to obtain broad coverage of different types: sports, technology, cultures, and general interests. These topics are shown in Table 2. We group the posting documents with the same hashtag '#' into clusters, and then we remove the hashtags and compute its *perplexity* with respect to the current cluster, defined as

$$2^{-\frac{1}{N}\sum_{w_i\in V}\log P(w_i)}$$

Perplexity is actually an entropy based evaluation. In this sense, the lower perplexity within the same hashtag cluster, the better performance in purity the hashtag cluster would have.

5.3.2 Extrinsic Evaluation

In addition to the intrinsic perplexity-based measurements on hashtag clusters, we also evaluate the effectiveness of our smoothed language models on the tasks of microblog search. Here are a few more details about our experimental setups. For the retrieval task, to avoid the laborious work of building a test collection by hand, we focus our evaluation efforts on documents that contained at least one hashtag. Given the 10 topics mentioned above, we process all documents with hashtags as follows: first, the ground truth labels (i.e., the hashtags) are removed from the documents. We then use the hashtag terms as queries to search for relevant posting documents. The ones originally with the hashtag are regarded as relevant while others not. Note that, the retrieval performance under this experimental setting is to some extent a lower bound, since some of the retrieved documents could be false negative: they do not contain the hashtag but they are indeed relevant.

For the retrieval task, we return the results as a ranking list given a search *query*, and the ranking list is checked by examining the *relevant* documents. We measured ranking performance using the normalized Discounted Cumulative Gain (nDCG) [5].

$$nDCG(k) = \frac{1}{N_\Delta}\sum\nolimits_{|\Delta|}\frac{1}{Z_\Delta}\sum_{i=1}^{k}\frac{2^{r_i}-1}{\log(1+i)}$$

where N_Δ denotes the total numbers of queries or users (Δ=q for queries), k indicates the top-k positions in a ranked list, and Z_Δ is a normalization factor obtained from a perfect ranking for a particular query. r_i is the judge score (i.e., 1: relevant/reposted, 0: irrelevant/unreposted) for the i-th posting document in the ranking list for the query.

Topic	EN-1	EN-2	EN-3	EN-4	EN-5	CN-1	CN-2	CN-3	CN-4	CN-5
LM	15851	11356	10676	7584	8257	22306	17441	10204	16887	9237
Additive	15195	10035	10342	7198	7924	19139	16221	10108	16342	9003
Absolute	15323	10123	10379	7230	8093	19403	16932	9984	16681	9111
Jelinek-Mercer	14115	10011	10185	9818	8003	20025	16201	10049	16001	8728
Dirichlet	13892	9516	10138	7124	7345	19712	16361	9119	15886	8550
PLM	13730	9925	10426	6913	7512	19965	15230	9865	14219	8981
CBDM	12931	9845	9311	6893	7510	19129	15194	9323	15113	7906
DELM	11853	9820	9513	7133	7348	18809	14165	9510	13985	7621
CSPLM	11306	9611	9105	6229	7155	19045	15887	9238	13953	7139
SLMS	10788	9539	8408	5817*	7109*	18169	15375	9194	13212	6919
SRS	11528	9712	9237	6618	7185	18947	16031	9113*	14029	7001
BSLMS	10294*	9376*	8361*	5985	7123	18012*	14923*	9204	13110*	6851*

Table 3: Perplexity of language models under different hashtag clusters. '$\star$' indicates that we accept the improvement hypothesis of BSLMS over the best rival baseline by Wilcoxon test at a significance level of 0.01.

	nDCG@5	nDCG@25	nDCG@50	MAP
LM	0.271	0.298	0.319	0.328
Additive	0.295	0.320	0.331	0.385
Absolute	0.283	0.328	0.378	0.367
Jelinek-Mercer	0.331	0.376	0.361	0.503
Dirichlet	0.365	0.387	0.408	0.555
PLM	0.392	0.413	0.399	0.532
CBDM	0.388	0.397	0.426	0.546
DELM	0.404	0.438	0.489	0.566
CSPLM	0.415	0.467	0.474	0.513
SLMS	0.463	0.492*	0.503	0.600
SRS	0.424	0.452	0.497	0.480
BSLMS	0.473*	0.490	0.509*	0.605*

Table 4: Retrieval performance against baselines. '$\star$' indicates that we accept the improvement hypothesis of TSLMS over the best baseline by Wilcoxon test at a significance level of 0.01.

We also evaluate the system in terms of Mean Average Precision (MAP) [16] under a similar judge assumption as above:

$$MAP = \frac{1}{N_\triangle}\sum\nolimits_{|\triangle|}\frac{1}{Z_\triangle}\sum_{i=1}^{k}P_i \times r_i$$

Here $N_\triangle$ is the number of documents associated with the query, $Z_\triangle$ is the number of relevant documents retrieved, and P_i is the precision at i-th position for the query.

5.4 Overall Performance

We compare the performance of all methods of language model smoothing in the two datasets, measured in the intrinsic evaluation of perplexity, as well as the extrinsic evaluation of retrieval. Table 3-5 list the overall results against all baseline methods. Our proposed method BSLMS shows clearly better performance than the baseline methods.

In general, compared with the textual information based language model smoothing, the advantage of our proposed method mainly comes from social information through the documents on social networks. We use a myriad of attribute factors and dependencies to control the influence propagation on the factor graph, and more importantly, we distinguish influence directions to make a more reliable estimation. Moreover, compared with other socialized language model smoothing methods, we characterize social influence propagation on bi-directional dimensions, which would be able to model the smoothing information in a finer-grained granularity.

Language model without any smoothing performs worst as expected, and once again demonstrates the severe weakness of data sparsity on social networks. Simple intuition based methods such as additive smoothing does not help a lot, since it only arbitrarily modifies the given term counts straightforward to avoid zero occurrence, which is proved to be insufficient. Absolute smoothing has a comparable performance as additive, due to the similar idea to reduce term counts naively. Jelinek-Mercer and Dirichlet methods are more useful since they include the information from the whole collection as background language models, but they fail to distinguish documents from documents and use all of them equally into smoothing. PLM offers a strengthened language model smoothing strategy within each posting document based on positions, and smooth the terms outside of the posting document formulating the background collection into a Dirichlet prior. The performance of CBDM and DELM indicates a prominent improvement, and proves that semantic attributes included into the smoothing process really make a difference. Both of the smoothing methods cluster documents, and use the clustered documents as a better background. However, none of these methods has made use of the social factors during the language model smoothing, while our method suggests social attributes, such as interactions and relationships, do have an impact on texts through influence propagation.

Among the baseline group of language model smoothing methods with social information, none of the proposed approaches incorporates direction information into the influence propagation and smoothing estimation. In this sense, the propagated influence is literally integrated as a whole while in fact, the influence should have directions. Influence + indicates impacts from the other posting documents and influence - means impacts on the others. With directions distinguished for influence propagation, we would be able to estimate the influence from source language model to target language model more accurately.

Having demonstrated the effectiveness of our proposed BSLMS, we carry the next move to investigate more analysis on parameter settings, and factor contributions.

5.5 Analysis and Discussions

5.5.1 Parameter Settings

In the experiments, as we use data from two consecutive months, we learn parameters θ=$\{\alpha, \beta\}$ on the data from the first month, and examine the performance on the testing data from the next month.

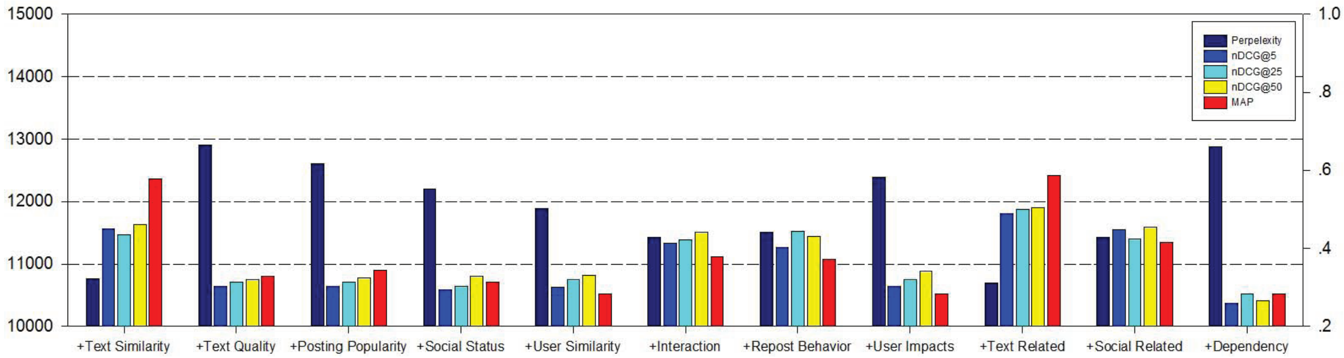

Figure 2: Performance comparison measured in perplexity, nDCG@25, and MAP in hashtag clustering and retrieval tasks for feature analysis. "+factor(s)" means the performance of individual factor (group) in isolation.

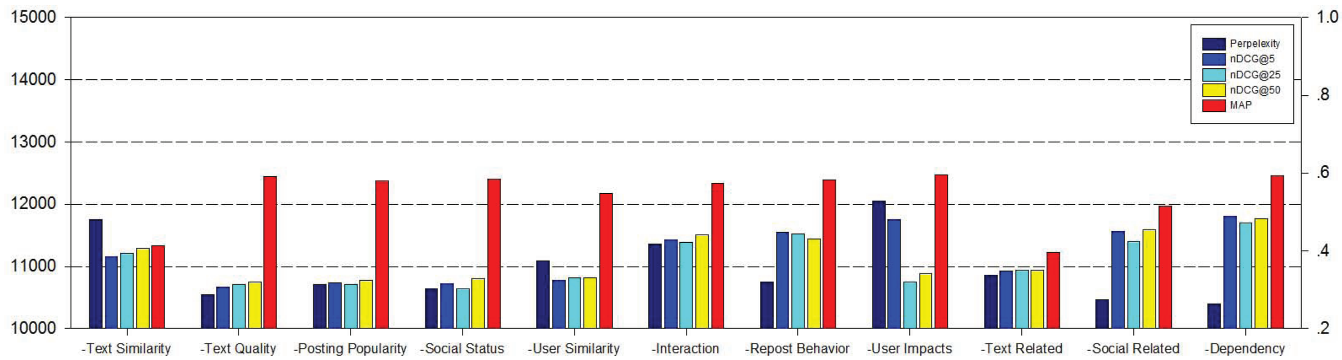

Figure 3: Performance comparison measured in perplexity, nDCG@25, and MAP in hashtag clustering and retrieval tasks. "-factor(s)" means the performance of individual factor (group) when dropped out from the all-feature model.

There is another free parameter λ in Equation (15) to balance the original language model and the smoothing language model. As we opt for more or less generic parameter value as we do not want to tune our method too much to suit the specific datasets at hand, we experiment with value ranging from 0 to 0.9, with a step size of 0.1. By examining the performance of optimum perplexity performance achieved on average for the two datasets, we set λ as 0.3.

5.5.2 Factor Contributions

We further analyze the contributions of the factors. We conduct to a detailed experiment on all separate factors and visualize the result in Figure 2-3. In the factor graph for socialized language model smoothing, we consider 9 different attributes and factors: (i) text similarity, (ii) text quality, (iii) posting popularity, (iv) social status, (v) user similarity, (vi) social interactions, (vii) repost behavior, (viii) user impacts and (ix) variable dependency. Besides, we combine factors (i)-(iii) as *text related* ones and (iv)-(viii) as *social related* ones. We also list the performance of BSLMS which employs all components here for comparison. Here we examine the contribution of the different factors defined in our model. To be specific, we show the performance of all the factors in isolation and then leave-one-out from the full combination of all features, one at a time.

From Figure 2 and 3, we see that all of the individual factors have positive contributions to our evaluation tasks. The first result in Figure 2 is performed using the correspond component only and the second group of results in Figure 3 is performed using the full factor combination exempting the corresponding component, using a leave-one-out manner. For the individual factor analysis, we could see that on average *text similarity* still contributes most in isolation and its absence leads to unfavorable decrease. As to the social related features, *interaction* is the most important social factor for measuring the propagated influence, and gets a clear drop on the performance when left out from full factor combination. It is natural to see through the reposting behavior, the language model for a particular user is influenced by others. We also examine the three aspects of feature groups, i.e., text related factors, social related factors and variable dependencies. Text related factors are proved to be more useful while the social group yields better performance when integrate the factors together. Dependency factor seems to be the least powerful predictor. It is understandable that dependency factor is not deterministic but just to balance the label values. In general, the combination of all factors will be beneficial to improve the performance, which indicates that our method works well by combining the different factor functions and each factor in our method contributes to the overall improvements.

6. CONCLUSIONS

We present a bi-directional influence propagation based language model smoothing method to solve the zero count phenomenon for online social networks. The social influence is estimated based on a factor graph model, by utilizing a series of attributes and dependency factors from both textual and social dimensions with direction information. In this way, we propagate the term occurrence along the networks with discounted term counts according to the estimated pairwise influence between documents, and finally smooth the sparse language model.

We examine the effect of our proposed language model smoothing method on a series of intrinsic and extrinsic evaluation metrics based on the Twitter dataset (in English) and Weibo dataset (in Chinese). Our proposed method consistently and significantly outperforms the alternative baselines: socialized language model smoothing with bi-directions outperforms that without such information. Furthermore, we have investigated factor contributions. In general, the features are demonstrated as effective, while direction information of influence further facilitates the factors. In the future, we will include more flexible social factors and make our model adaptive to diversified online social networks.

7. ACKNOWLEDGMENTS

This work is partially supported by the major project of National Social Science Fund 12&2D223, National Natural Science Foundation of China (NSFC) Grant No. 61532008 and Grant No. 61402191. We thank all the anonymous reviewers for their valuable and constructive comments.

8. REFERENCES

[1] P. Analytics. Twitter study–august 2009. 15, 2009.

[2] K. Chen, T. Chen, G. Zheng, O. Jin, E. Yao, and Y. Yu. Collaborative personalized tweet recommendation. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '12, pages 661–670, New York, NY, USA, 2012. ACM.

[3] J. Hopcroft, T. Lou, and J. Tang. Who will follow you back?: Reciprocal relationship prediction. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 1137–1146, New York, NY, USA, 2011. ACM.

[4] Y.-Y. Huang, R. Yan, T.-T. Kuo, and S.-D. Lin. Enriching cold start personalized language model using social network information. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, ACL'14, pages 611–617, Stroudsburg, PA, USA, 2014. Association for Computational Linguistics.

[5] K. Järvelin and J. Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, Oct. 2002.

[6] R. Kindermann, J. L. Snell, et al. *Markov random fields and their applications*, volume 1. American Mathematical Society Providence, RI, 1980.

[7] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on*, 47(2):498–519, 2001.

[8] T.-T. Kuo, R. Yan, Y.-Y. Huang, P.-H. Kung, and S.-D. Lin. Unsupervised link prediction using aggregative statistics on heterogeneous social networks. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 775–783, New York, NY, USA, 2013. ACM.

[9] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 591–600, New York, NY, USA, 2010. ACM.

[10] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 111–119, New York, NY, USA, 2001. ACM.

[11] V. Lavrenko and W. B. Croft. Relevance based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[12] C. Lin, C. Lin, J. Li, D. Wang, Y. Chen, and T. Li. Generating event storylines from microblogs. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 175–184, New York, NY, USA, 2012. ACM.

[13] J. Lin, R. Snow, and W. Morgan. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 422–429, New York, NY, USA, 2011. ACM.

[14] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '04, pages 186–193, New York, NY, USA, 2004. ACM.

[15] Y. Lv and C. Zhai. Positional language models for information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 299–306, New York, NY, USA, 2009. ACM.

[16] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. 2008.

[17] Q. Mei, D. Zhang, and C. Zhai. A general optimization framework for smoothing language models on graph structures. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 611–618, New York, NY, USA, 2008. ACM.

[18] H. Ney, U. Essen, and R. Kneser. On the estimation ofsmall'probabilities by leaving-one-out. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(12):1202–1212, 1995.

[19] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[20] E. Pitler, A. Louis, and A. Nenkova. Automatic evaluation of linguistic quality in multi-document summarization. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 544–554, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[21] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 275–281, New York, NY, USA, 1998. ACM.

[22] F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, CIKM '99, pages 316–321, New York, NY, USA, 1999. ACM.

[23] J. Tang, T. Lou, and J. Kleinberg. Inferring social ties across heterogenous networks. In *Proceedings of the Fifth ACM International Conference on Web Search and Data Mining*,

WSDM '12, pages 743–752, New York, NY, USA, 2012. ACM.

[24] J. Tang, J. Sun, C. Wang, and Z. Yang. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 807–816, New York, NY, USA, 2009. ACM.

[25] T. Tao, X. Wang, Q. Mei, and C. Zhai. Language model information retrieval with document expansion. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 407–414, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[26] J. Teevan, D. Ramage, and M. R. Morris. #twittersearch: A comparison of microblog search and web search. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 35–44, New York, NY, USA, 2011. ACM.

[27] Z. Wang, J. Li, Z. Wang, and J. Tang. Cross-lingual knowledge linking across wiki knowledge bases. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 459–468, New York, NY, USA, 2012. ACM.

[28] S. Wu, J. Sun, and J. Tang. Patent partner recommendation in enterprise social networks. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 43–52, New York, NY, USA, 2013. ACM.

[29] R. Yan, H. Jiang, M. Lapata, S.-D. Lin, X. Lv, and X. Li. Semantic v.s. positions: Utilizing balanced proximity in language model smoothing for information retrieval. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*, IJCNLP'13, pages 507–515, 2013.

[30] R. Yan, L. Kong, C. Huang, X. Wan, X. Li, and Y. Zhang. Timeline generation through evolutionary trans-temporal summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 433–443, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[31] R. Yan, M. Lapata, and X. Li. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 516–525, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[32] R. Yan, X. Li, M. Liu, and X. Hu. Tackling sparsity, the achilles heel of social networks: Influence propagation based language model smoothing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, ACL-IJCNLP'15, pages 623–629, Stroudsburg, PA, USA, 2015. Association for Computational Linguistics.

[33] R. Yan, X. Wan, J. Otterbacher, L. Kong, X. Li, and Y. Zhang. Evolutionary timeline summarization: A balanced optimization framework via iterative substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 745–754, New York, NY, USA, 2011. ACM.

[34] R. Yan, I. E. Yen, C.-T. Li, S. Zhao, and X. Hu. Tackling the achilles heel of social networks: Influence propagation based language model smoothing. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, pages 1318–1328, Republic and Canton of Geneva, Switzerland, 2015. International World Wide Web Conferences Steering Committee.

[35] Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li. Social context summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 255–264, New York, NY, USA, 2011. ACM.

[36] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '01, pages 334–342, New York, NY, USA, 2001. ACM.

[37] J. Zhao and Y. Yun. A proximity language model for information retrieval. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 291–298, New York, NY, USA, 2009. ACM.

[38] W. X. Zhao, J. Jiang, J. Weng, J. He, E.-P. Lim, H. Yan, and X. Li. Comparing twitter and traditional media using topic models. In *Proceedings of the 33rd European Conference on Information Retrieval*, ECIR '11, pages 338–349. 2011.

[39] X. W. Zhao, Y. Guo, R. Yan, Y. He, and X. Li. Timeline generation with social attention. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '13, pages 1061–1064, New York, NY, USA, 2013. ACM.