Open Domain Short Text Conceptualization: A Generative + Descriptive Modeling Approach

Yangqiu Song[a] **Shusen Wang**[b] **Haixun Wang**[c]
[a]University of Illinois at Urbana-Champaign [b]Zhejiang University [c]Google Research
[a]yqsong@illinois.edu [b]wss@zju.edu.cn [c]haixun@google.com

Abstract

Concepts embody the knowledge to facilitate our cognitive processes of learning. Mapping short texts to a large set of open domain concepts has gained many successful applications. In this paper, we unify the existing conceptualization methods from a Bayesian perspective, and discuss the three modeling approaches: descriptive, generative, and discriminative models. Motivated by the discussion of their advantages and shortcomings, we develop a generative + descriptive modeling approach. Our model considers term relatedness in the context, and will result in disambiguated conceptualization. We show the results of short text clustering using a news title data set and a Twitter message data set, and demonstrate the effectiveness of the developed approach compared with the state-of-the-art conceptualization and topic modeling approaches.

1 Introduction

Short text conceptualization is a task to map a piece of short text to a large set of open domain concepts with different granularities.[1] Since short texts are usually lack of context, mapping short texts to concepts can help better make sense of text data, extend the texts with categorical or topical information, and facilitate many applications. For example, it has been verified very useful for word/phrase similarity/relatedness measure [Gabrilovich and Markovitch, 2007; Li *et al.*, 2013; Agrawal *et al.*, 2014], short text categorization [Gabrilovich and Markovitch, 2006; Wang *et al.*, 2014], Twitter messages clustering [Song *et al.*, 2011], search relevance measurement [Egozi *et al.*, 2011; Song *et al.*, 2014], search log mining [Hua *et al.*, 2013], advertising keywords semantic matching [Liu *et al.*, 2012; Kim *et al.*, 2013], and dataless text classification by label understanding [Chang *et al.*, 2008; Song and Roth, 2014; 2015].

[1]In this paper, we focus on the explicit concept mapping approaches. For more comparisons of explicit and latent semantic analysis for text representation, please refer to [Huang *et al.*, 2012; Song and Roth, 2014] for more details.

Typical concept mapping methodologies include the so called probabilistic conceptualization [Song *et al.*, 2011] and explicit semantic analysis (ESA) [Gabrilovich and Markovitch, 2009]. We first briefly review the two models as follows.

Probabilistic conceptualization: Given a set of terms (words or multiple-word expressions) $E = \{e_1, \ldots, e_M\}$ in a short text[2], probabilistic conceptualization tries to find the concepts associated with scores that can best describe the terms. Suppose we have a general and open domain concept set $C = \{c_1, \ldots, c_T\}$. In probabilistic conceptualization, it makes the naive Bayes assumption of the conditional probabilities and uses

$$P(c_t|E) = P(E|c_t)P(c_t)/P(E) \propto P(c_t)\prod_{m=1}^{M} P(e_m|c_t) \quad (1)$$

as the score associated with c_t. Here, $P(e_m|c_t) = \frac{n(e_m,c_t)}{n(c_t)}$ where $n(e_m, c_t)$ is the co-occurrence frequency of concept c_t and term e_m in the sentences used by information extraction, and $n(c_t)$ is the overall number of concept c_t. Moreover, $P(c_t) = \frac{n(c_t)}{\sum_t n(c_t)}$ is normalized by the number of all the concepts in C. The basic assumption behind this model is that given each concept c_t, all the observed terms $e_m \in E$ are conditionally independent. Then it uses the probability $P(c_t|E)$ to rank the concepts and selects the concepts with the largest probabilities to represent the text containing the terms in E. However, this has a major drawback:

- Naive Bayes will quickly boost the concepts co-occurred with all the observed terms in the short text due to the multiplication term $\prod_{m=1}^{M} P(e_m|c_t)$, and dismiss the concepts partially matching the terms. In particular, in some extreme cases, only the general and vague concepts, e.g., *topic* or *thing*, can be retrieved co-occurring with all the terms, whereas, the partially matched concepts would be more specific and descriptive to represent the text.

Explicit semantic analysis (ESA): ESA simply combines the weighted concepts of each term in a short text. We use $\mathbf{e}_m = (e_{m,1}, ..., e_{m,T}) \in \mathbb{R}_+^T$ to represent the concept vector

[2]Parsing short text to be words or multi-word expressions can be non-trivial [Song *et al.*, 2014]. We ignore this since it is not the focus of this paper.

Table 1: A comparison of the union and intersection methods.

	apple and microsoft	*obama's real-estate policy*
Intersection	company, brand, manufacturer, ...	**topic, thing, issue, term, example**, ...
Union	company, brand, manufacturer, **fruit, juice**, ...	president, politician property, asset, plan, ...

of the term e_m. For example, we can set $e_{m,t} = f(n(e_m, c_t))$ as a function of the co-occurrence of the term e_m and c_t. In the original ESA, it uses TF-IDF (term frequency-inverse document frequency) score of e_m shown in the t-th Wikipedia page, which is denoted as a concept c_t. We use a vector $\mathbf{c} = (c_1, ..., c_T) \in \mathbb{R}_+^T$ to denote the concept proportion that can describe the whole short text containing $E = \{e_1, \ldots, e_M\}$. Then ESA recalls the concepts with scores as this:

$$\mathbf{c} = \sum\nolimits_{m=1}^{M} w_m \mathbf{e}_m, \tag{2}$$

where w_m is the weight associated to e_m, e.g., the TF-IDF score of e_m in the short text. The benefit of using this representation is that the values in the concept vectors $\mathbf{e}_m$ are not restricted to the co-occurrence frequencies, but can be arbitrarily tuned. However, it is still not without problems:

- The resulting concept vectors can be noisy. For example, for the text "microsoft unveils office for apple's ipad," we all know that in this context "apple" should not be a *fruit*. However, simply adding $\mathbf{e}_m$ will also introduce *fruit* as a concept to describe the text. The backend intuition of this computation is that it assumes that there is only one term cluster in the short text, and uses the (weighted) mean of concept vectors, which is the center of the terms in concept vector space, to represent the text, regardless the sense of the word. Particularly, sense disambiguation is more serious for short texts such as tweets and search queries, since with more words, the impact of the ambiguous concepts will be reduced as less significant.

We can use two operations to illustrate the results of probabilistic conceptualization and ESA: *intersection* used by probabilistic conceptualization and *union* used by ESA. In Table 1 we see that intersection of concepts for "obama" and "real-estate policy" will get *topic*, *thing*, *issue*, etc., while union of the concepts for "apple" and "microsoft" will have concepts such as *fruit* but not correct to represent their meaning. Thus, intersection of different concept sets will sharpen the meaning of the representation, while union will broaden the meaning. When the terms in a short text are related, intersecting the concepts can help us disambiguate them. However, when the terms are not related, intersection will get only very general or vague concepts.

Given the above analysis that both approaches are with modeling shortcomings for short text conceptualization, in this paper we propose an approach that can incorporate both intersection and union operations. The contributions of this paper can be summarized as follows.

- We show how existing conceptualization approaches can be reformulated as descriptive, generative and discriminative models in a framework. This is the first attempt to unify different short text conceptualization methods.
- We introduce a *generative + descriptive* modeling approach under the framework for short text conceptualization and demonstrate its effectiveness using a news title data set and a Twitter message data set in the experiments.

2 Descriptive, Generative, and Discriminative Modeling

To summarize from the modeling perspective, analogous to the image conceptualization frameworks discussed in [Zhu, 2003], we also introduce and analyze three ways to perform short text conceptualization as: descriptive, generative and discriminative models. In the descriptive and generative models, we consider to model the probability $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$. In the discriminative model, we consider directly modeling the probability $P(\mathbf{c}|\mathbf{e}_1, ..., \mathbf{e}_M)$.

Descriptive Model (Causal Markov Model): The probabilistic conceptualization can be regard as a simple causal Markov model, since it imposes the partial order of the probabilities of concept-term relationship. We first assume the conditional independency of $\mathbf{e}_m$ given $\mathbf{c}$: $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c}) = \Pi_m^M P(\mathbf{e}_m|\mathbf{c})$. Then we define $P(\mathbf{e}_m|\mathbf{c}) \propto \Pi_t^T P(e_{m,t}|P(e_m|c_t)) = \Pi_t^T P(e_m|c_t)^{e_{m,t}}$ as a multinomial distribution where $P(e_m|c_t)$ is calculated based on the evidence of co-occurrence in knowledge base (explained under Eq. (1)). We define $e_{m,t} = 1$ if for this trial c_t is selected as the description of the short text and $e_{m,t'} = 0$ for $t' \neq t$. Now we can factorize $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$ as:

$$P(e_1|c_1)^{e_{1,1}} \cdot ... \cdot P(e_1|c_T)^{e_{1,T}} \cdot ... \cdot P(e_M|c_T)^{e_{M,T}}, \tag{3}$$

By incorporating the prior $P(\mathbf{c}) \triangleq \prod_{t=1}^T P(c_t)$, we can re-write the posterior of $\mathbf{c}$:

$$\begin{aligned} P(\mathbf{c}|\mathbf{e}_1, ..., \mathbf{e}_M) &\propto P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})P(\mathbf{c}) \\ &= \prod\nolimits_{t=1}^{T} P(c_t) \prod\nolimits_{m=1}^{M} P(e_m|c_t)^{e_{m,t}}. \end{aligned} \tag{4}$$

Then selecting the top k concepts using Eq. (1) among all the T concepts can be considered as the maximum a posterior (MAP) estimation of this posterior in Eq. (4). This illustrates what probabilistic conceptualization really optimizes. Thus, if one of the probability $P(e_m|c_t)$ equals to zero, then the whole probability $P(\mathbf{c}|\mathbf{e}_1, ..., \mathbf{e}_M)$ equals to zero. Even if a smoothing technique can be applied [Song *et al.*, 2011], the probability mass $P(\mathbf{c}|\mathbf{e}_1, ..., \mathbf{e}_M)$ could be too small to be reasonable in this case.

Generative Model: ESA can be regarded as a generative model since it uses the concept-term relationship as the evidence of generated features of terms, and estimates the latent concept distribution which generates the features. If we formulate the probability $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$ as:

$$\begin{aligned} P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c}) &= \prod\nolimits_{m=1}^{M} P(\mathbf{e}_m|\mathbf{c}) \\ &\propto \prod\nolimits_{m=1}^{M} \exp\{-||\mathbf{e}_m - \mathbf{c}||^2\}, \end{aligned} \tag{5}$$

where $P(\mathbf{e}_m|\mathbf{c})$ is assumed to be a Gaussian distribution centered by the underlying concept distribution $\mathbf{c}$. Then $\mathbf{c} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{e}_m$ is the maximum likelihood estimate with the probability $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$. Here $P(\mathbf{e}_m|\mathbf{c})$ is more flexible and not necessarily to be factorized as $\Pi_t^T P(e_m|c_t)$. For example, $e_{m,t}$ $(t = 1, ..., T)$ in the concept vector $\mathbf{e}_m$ can be the co-occurrence frequency of concept c_t and term e_m in the same sentence or same document. We can also define $e_{m,t} \triangleq P(c_t|e_m)$ which is the typicality of a concept c_t to describe the term e_m, or $P(e_m|c_t)$, which is the typicality of how much a term e_m can instantiate the concept c_t.

The formulation in Eq. (5) also explains why explicit semantic analysis assumes that there is only one cluster of the terms observed in the short text. A natural way to extend this is to perform clustering by assuming there are multiple clusters of concept vectors. However, there is still problem if we do not consider the concept intersection problem inside term clusters, since the computation of a cluster center is the average of all the vectors to represent the terms inside the cluster. In this case, the ambiguous concepts will still show up in the final representation.

Discriminative Model: Yet another way for conceptualization is to classify the short text onto a predefined taxonomy or ontology. Classification can be regarded as the discriminative model which wants to estimate $\mathbf{c}$ by directly modeling the probability $P(\mathbf{c}|\mathbf{e}_1, ..., \mathbf{e}_M)$. For example, we can learn (or simply find) a set of projection vectors $\mathbf{w}_t, t = 1, ..., T$, to project the observed text to maximize $P(c_t|\mathbf{w}_t, \mathbf{e}_1, ..., \mathbf{e}_M) = \frac{1}{Z}f(\mathbf{w}_t, g(\mathbf{e}_1, ..., \mathbf{e}_M))$, where the concept vector is considered as a feature vector to generate the representation of the short text. A typical $g(\mathbf{e}_1, ..., \mathbf{e}_M)$ can be $\frac{1}{M}\sum_{i=m}^{M}\mathbf{e}_m$ (more representations can be found in [Song and Roth, 2014]). Since discriminative model is costly when the number of concepts is large (e.g., millions of concepts) and thus is not the focus of this paper, we do not expand this direction and leave for further development and comparison.

We can see that both the simple descriptive and generative approaches factorize the probability as $\prod_{m=1}^{M} P(\mathbf{e}_m|\mathbf{c})$, which do not consider the relationships between e_m's. In the following section we introduce a generative + descriptive model that tries to jointly model $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$ to incorporate the relationships between terms with more descriptive power.

3 Generative + Descriptive Conceptualization

In this section, we introduce our generative + descriptive conceptualization model. We incorporate the term relationships into the generative model, and formulate it as a Markov random field (MRF). Then we regard conceptualization as the latent variable inference problem of the MRF model.

3.1 Graphical Model

Since terms can be used to disambiguate each other if they have relationships, we want to break the i.i.d. assumption used by the above descriptive or generative models which factorize the conditional joint probability as $\prod_{m=1}^{M} P(\mathbf{e}_m|\mathbf{c})$. We introduce a graph built on the terms $E = \{e_1, \ldots, e_M\}$ and introduce an energy function for each maximal cliques in the graph.

Intuitively, if a short text contains both "apple" and "microsoft," then the importance of concept "company" will be larger and the concept "fruit" is not an appropriate concept to describe both terms. We introduce the probability of $P(concept\ vector\ of\ \{apple, microsoft\}|\mathbf{c})$ to remove the ambiguity. Particularly, we represent the feature of tth concept related to "apple" and "microsoft" as $\bar{I}_0(e_{apple,t}) \cdot \bar{I}_0(e_{microsoft,t}) \cdot (e_{apple,t} + e_{microsoft,t})$, where $\bar{I}_0(x) = 1$ if $x \neq 0$ and $\bar{I}_0(x) = 0$ if $x = 0$. In this case, only their common concepts are considered. The common concept detection for related terms then corresponds to the intersection mechanism.

Formally, to introduce the relationship between observed terms in a generative model, we build an undirected graph to describe them, and factorize the joint probability based on its maximal cliques. An example graphical model is shown in Figure 1(a). If we have parsed terms $e_1, \ldots, e_M$ in a short text, and detected the relationships between e_1, e_2, and e_3, then in the graphical model, we have a maximal clique: $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$. In this case, instead of mapping the single terms to concepts, we map the cliques to concepts. We also denote $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_T) \in \mathbb{R}_+^T$ as the hyperparameter of the prior of the concept distribution $\mathbf{c}$. In the following, we first show how to formulate the concept vectors $\mathbf{e}_m$'s, and then show how to parameterize the joint probability $P_\Phi(\boldsymbol{\alpha}, \mathbf{c}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M)$.

Concept Vector for Each Term

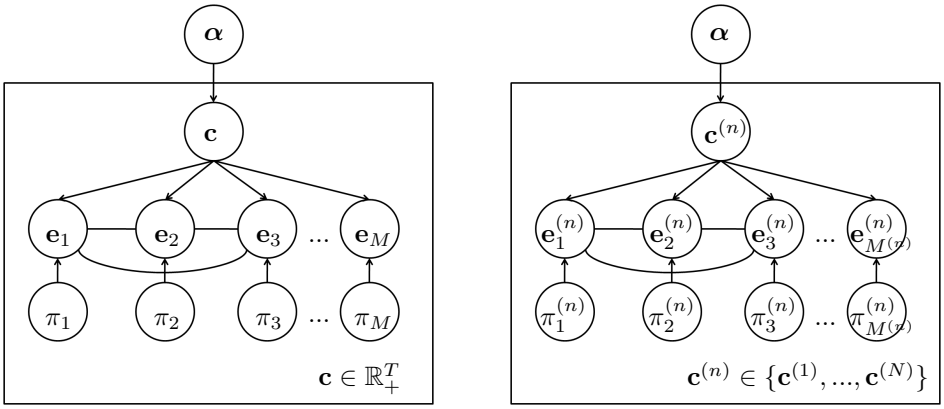
We use Probase [Wu *et al.*, 2012] here as the knowledge base to demonstrate the conceptualization framework. Probase uses an automatic and iterative procedure to extract concept knowledge from 1.68 billion Web pages [Wu *et al.*, 2012]. It contains 2.36 millions of open domain concepts, and provides around 14 millions relationships with two kinds of important knowledge related to concepts: concept-attribute co-occurrence and concept-instance co-occurrence.[3] When we detect a term e_m in a short text, we introduce a type indicator π_m to indicate whether e_m is an attribute ($\pi_m = 0$) or an instance ($\pi_m = 1$). Then the concept vector $\mathbf{e}_m \in \mathbb{Z}_+^T$ representing e_m is defined as:

$$\mathbf{e}_m = \begin{cases} A_{\cdot,e_m} & \text{if } \pi_m = 0 \\ B_{\cdot,e_m} & \text{if } \pi_m = 1 \end{cases} . \quad (6)$$

We denote $\mathbf{A} \in \mathbb{Z}_+^{T\times V}$ is the concept-attribute co-occurrence matrix, where V is the number of distinctive instances and attributes in the knowledge base. The (t, v)-th entry $A_{t,v}$ is an integer representing the co-occurrence count of concept t and attribute v, and $A_{\cdot,e_m}$ is the e_m's column of $\mathbf{A}$. Similarly, $\mathbf{B} \in \mathbb{Z}_+^{T\times V}$ is the concept-instance co-occurrence matrix.

Ideally, the graphical model is a mixture model and $\{\pi_m\}_{m=1}^M$ should be regarded as hidden variables. We need to apply the expectation-maximization (EM) algorithm to infer $\{\pi_m\}_{m=1}^M$ and combine $A_{\cdot,e_m}$ and $B_{\cdot,e_m}$ to generate $\mathbf{e}_m$. However, considering that there are only less than 0.1% terms

[3]The data are available at http://probase.msra.cn/.



(a) Conceptualization of one short text. (b) Corpus adaptation.

Figure 1: Partially directed graphical models for short text conceptualization. $\mathbf{e}_m$ and $\mathbf{e}_m^{(n)}$ represent the concept vector of a term. $\mathbf{c}$ and $\mathbf{c}^{(n)}$ represent the concept distribution to generate the short text. $\boldsymbol{\alpha}$ is the hyper-parameter.

in Probase which can act as both instances and attributes, it is not worth using EM in most of the cases. We therefore use the following heuristic rules to determine a term's concept vector: (i) *attribute seldom appears alone*: if a term is not related to any other terms, it is invariably an instance; (ii) *mutually exclusive*: if a term acts as an attribute in a sentence, then it cannot act as an instance simultaneously. Then we compute $\{\pi_m\}_{m=1}^M$ in advance and simply treat them as observed variables. In the following sub-section, we will show how to determine the relationships between terms.

Clique Detection

We define $r^{(\text{i–i})}(e_i, e_j)$ to measure the strength of the instance–instance relationship, which is defined as the cosine similarity between two vectors $B_{\cdot,e_i}$ and $B_{\cdot,e_j}$: $r^{(\text{i–i})}(e_i, e_j) = \frac{B_{\cdot,e_i}^T B_{\cdot,e_j}}{||B_{\cdot,e_i}|| \cdot ||B_{\cdot,e_j}||}$. The strength of the instance–attribute relationship $r^{(\text{i–a})}(e_i, e_j)$ is similarly defined by using $B_{\cdot,e_i}$ and $A_{\cdot,e_j}$. Note that other metrics or data sources to compute the term relatedness can be applied [Gabrilovich and Markovitch, 2007; Li *et al.*, 2013; Huang *et al.*, 2012]. Here we only use this simplest implementation to demonstrate the framework. Given a tolerance τ, an edge between $\mathbf{e}_i$ and $\mathbf{e}_j$ is introduced if

$$r(e_i, e_j) \triangleq \max\left\{r^{(\text{i–i})}(e_i, e_j), r^{(\text{i–a})}(e_i, e_j), r^{(\text{i–a})}(e_j, e_i)\right\} \geq \tau; \tag{7}$$

otherwise e_i and e_j are not linked. For example, if both "apple" and "microsoft" appear in the short text, we will build an edge between them since the similarity is large. Another example is if "population" co-occurs with "new york city," then population is regarded as an attribute since the concept vector for attribute "population" (concepts are *country*, *city*, *location*, *region*, etc.) has much larger similarity than the concept vector of instance "population" (concepts are *geographical data*, *data*, *information*, etc.) with the concept vector of "new york city."

Factorization

Suppose there are K maximal cliques, which cannot be extended by including any one more adjacent $\mathbf{e}_m$ representing e_m. Let $\mathcal{I}_k$ be the set of indices of those terms in the k-th maximal clique, and $\mathcal{E}_k = \{\mathbf{e}_m, \pi_m\}_{m \in \mathcal{I}_k}$. Then $\mathcal{E}_k \cup \{\mathbf{c}\}$ is a maximal clique of the moralized graph [Koller and Friedman, 2009]. We factorize the joint distribution as:

$$P_\Phi(\boldsymbol{\alpha}, \mathbf{c}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M) = \frac{1}{Z}\, \phi(\boldsymbol{\alpha}, \mathbf{c}) \prod_{k=1}^K \phi(\mathcal{E}_k, \mathbf{c}), \tag{8}$$

where Z is the partition function. We denote $\mathbf{f}(\mathcal{E}_k) \in \mathbb{Z}_+^T$ be the feature vector of the clique which has a multinomial distribution parameterized by $\mathbf{c}$ and $f_t(\mathcal{E}_k)$ is the tth entry of $\mathbf{f}(\mathcal{E}_k)$. Then the factor $\phi(\mathcal{E}_k, \mathbf{c})$ is defined as

$$\phi\Big(\mathcal{E}_k, \mathbf{c}\Big) = \prod_{t=1}^T c_t^{f_t(\mathcal{E}_k)} \tag{9}$$

where $f_t(\mathcal{E}_k) = (\prod_{i \in \mathcal{I}_k} \bar{I}_0(e_{i,t})) \cdot (\sum_{j \in \mathcal{I}_k} e_{j,t})$, and $\bar{I}_0(x) = 1$ if $x \neq 0$; $\bar{I}_0(x) = 0$ if $x = 0$. The feature function $f_t(\mathcal{E}_k)$ sums the co-occurrence counts only if the related terms (i.e., in the same clique) all have this concept t; otherwise this concept is discarded. For example, if "apple" and "microsoft" appear together, concepts such as "fruit" will be filtered out.

Finally, we define a Dirichlet prior distribution parameterized by $\boldsymbol{\alpha}$ for the multinomial distribution parameter $\mathbf{c}$, i.e.,

$$\phi(\mathbf{c}, \boldsymbol{\alpha}) = P(\mathbf{c}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{t=1}^T \alpha_t)}{\prod_{t=1}^T \Gamma(\alpha_t)} \prod_{t=1}^T c_t^{\alpha_t - 1}, \tag{10}$$

which is a conjugate prior of multinomial distribution. If we have no prior knowledge of which concepts are more important, we can use symmetric $\boldsymbol{\alpha}$, i.e., all entries of $\boldsymbol{\alpha}$ are equal.

3.2 Latent Variable Inference: Conceptualization

Given the factorized joint probability distribution $P_\Phi(\boldsymbol{\alpha}, \mathbf{c}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M)$, we want to infer the latent variable $\mathbf{c}$ by the MAP estimation. Since $\mathbf{c}$ is modeled as a multinomial distribution to generate the concept vectors for the maximal cliques, we can then use the inferred concept distribution to describe short text. We can also call this procedure as a probabilistic conceptualization.

Given Eqs. (8), (9) and (10), the posteriori of $\mathbf{c}$ over the factors $\Phi = \{\phi(\boldsymbol{\alpha}, \mathbf{c})\} \bigcup \{\phi(\mathcal{E}_k, \mathbf{c})\}_{k=1}^K$ can be rewritten as

$$P_\Phi(\mathbf{c}|\boldsymbol{\alpha}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M)) \propto \prod_{t=1}^T c_t^{\alpha_t - 1 + \sum_{k=1}^K f_t(\mathcal{E}_k)}. \tag{11}$$

Given $\{\pi_m\}_{m=1}^M$ and $\boldsymbol{\alpha}$ fixed, and $\{\mathbf{e}_m\}_{m=1}^M$ determined, we maximize Eq. (11) w.r.t. $\mathbf{c}$, and the solution is:

$$c_t^{\text{opt}} = \frac{\alpha_t - 1 + \sum_{k=1}^K f_t(\mathcal{E}_k)}{\sum_{t=1}^T \Big(\alpha_t - 1 + \sum_{k=1}^K f_t(\mathcal{E}_k)\Big)}, \forall\, t = 1, \ldots, T. \quad (12)$$

As a special case of (12), when the terms are (assumed) independent, the solution is $c_t^{\text{opt}} = \Big(\alpha_t - 1 + \sum_{m=1}^M e_{m,t}\Big) / \sum_{t=1}^T \Big(\alpha_t - 1 + \sum_{m=1}^M e_{m,t}\Big)$, $\forall\ t = 1, \ldots T$. The solution $\mathbf{c}^{\text{opt}}$ has the following explanations. If some terms are related with each other, only their mutual concepts are summed. If all the terms are independent, the concept distribution is proportional to the sum of the co-occurrence count of the concept and each term plus the prior. This results in a similar solution as ESA. While ESA uses the maximium likelihood estimation, which relates to $P(\mathbf{e}_1, ..., \mathbf{e}_M|\mathbf{c})$, our solution uses MAP estimation, which relates to $P(\mathbf{c}|\boldsymbol{\alpha}, \mathbf{e}_1, ..., \mathbf{e}_M)$.

3.3 Hyperparameter Estimation: Corpus Adaptation

The Dirichlet prior of concept distribution $\mathbf{c}$ is parameterized by $\boldsymbol{\alpha}$. Larger α_t indicates that concept t is more important for all the short texts in a corpus. If the corpus is general, we can use symmetric $\boldsymbol{\alpha}$. When the corpus is of several specific topics such as "technology" and "business," some concepts such as "IT," "company" and "industry" are more common than the others. In this situation it is necessary to strengthen the important concepts by setting the corresponding entries of $\boldsymbol{\alpha}$ large. For this reason, we provide a maximum likelihood estimation method for learning the hyperparameter $\boldsymbol{\alpha}$ based on a corpus.

By integrating out $\mathbf{c}$ in $P_\Phi(\boldsymbol{\alpha}, \mathbf{c}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M)$, the resulting distribution over $\boldsymbol{\alpha}$ is

$$P(\boldsymbol{\alpha}, \{\mathbf{e}_m\}_{m=1}^M, \{\pi_m\}_{m=1}^M) \quad (13)$$
$$= \frac{\Gamma(\sum_{t=1}^T \alpha_t)}{\Gamma\Big(\sum_{t=1}^T (\alpha_t + \sum_{k=1}^K f_t(\mathcal{E}_k))\Big)} \prod_{t=1}^T \frac{\Gamma\Big(\alpha_t + \sum_{k=1}^K f_t(\mathcal{E}_k)\Big)}{\Gamma(\alpha_t)}.$$

As shown in Fig. 1(b), consider we are given N short texts; the m-th term parsed from the n-th text is denoted as $e_m^{(n)}$ ($\mathbf{e}_m^{(n)}$ as its concept vector), and ${\pi_m}^{(n)}$, $\mathcal{E}_k^{(n)}$ are similarly defined. Suppose the texts are i.i.d. with a common parameter $\boldsymbol{\alpha}$, the log likelihood function of the N texts is

$$\log P(\boldsymbol{\alpha}) \triangleq \sum_{n=1}^N \log P(\boldsymbol{\alpha}, \{\mathbf{e}_m^{(n)}\}_{m=1}^{M^{(n)}}, \{\pi_m^{(n)}\}_{m=1}^{M^{(n)}}). \quad (14)$$

The hyperparameter $\boldsymbol{\alpha}$ can be learned on the corpus of the N texts by the following fixed-point iteration:

$$\alpha_t^{\text{new}} \leftarrow \frac{\alpha_t \sum_{n=1}^N \Big(\Psi\big(\alpha_t + \sum_{k=1}^{K^{(n)}} f_t(\mathcal{E}_k^{(n)})\big) - \Psi(\alpha_t)\Big)}{\sum_{n=1}^N \Big(\Psi\big(\sum_{t=1}^T (\alpha_t + \sum_{k=1}^{K^{(n)}} f_t(\mathcal{E}_k^{(n)}))\big) - \Psi(\sum_{t=1}^T \alpha_t)\Big)}, \quad (15)$$

for $t = 1, \cdots, T$, where $\Psi(x) = d\log(\Gamma(x))/dx$ is the digamma function. The resulting $\boldsymbol{\alpha}^*$ maximizes the log likelihood function (14). The proof is shown in [Minka, 2003].

4 Experiments

In this section, we show experiments on two short text data sets to compare our method with existing conceptualization methods.

News Title: We extract news titles from a news corpus containing about one million articles searched from Web pages. The news articles have been classified into topics. We select six topics, i.e., company, disease, entertainment, food, politician, and sports, to evaluate different approaches. We randomly select 3,000 news articles in each topic, and only keep the title field. We call this data set the *News Title Data Set*. The average word count of the 18,000 news titles is 7.96.

Twitter: In this data set, the 4,542 tweets are in three categories: *company* (1,205), *country* (1,747), and *device* (1,590). The data in *company* category includes tweets about microsoft, google, apple, etc. The data in *country* category includes tweets about china, india, usa, japan, isreal, canada, etc. The data in *device* category includes tweets about kindle, iphone, xbox, etc. The average length of the Tweets is 13.36 words. Tweets are more noisy than news titles. For example, the tweets "Win an Amazon Kindle 3G Wireless from @FreeLunched Quick and easy registration at http://bit.ly/9fBuw4" and "Conker, Live and Reloaded - XBox game #xbox" have no overlapped terms, but they should be grouped together in this problem.

4.1 Methods and Settings

We first use each method to obtain concepts (or topics) of each short text in the two data sets. Then we use the concepts (or topics) as features and run spherical K-means clustering [Dhillon and Modha, 2001] to evaluate each method. To evaluate our method, we mainly compare it with the bag-of-words approach weighted by TF-IDF scores [Salton and McGill, 1983], LDA [Blei *et al.*, 2003], probabilistic conceptualization (Song *et al.* 's approach [Song *et al.*, 2011]), and ESA [Gabrilovich and Markovitch, 2007], since these approaches are most related to ours.

TF-IDF: TF-IDF represents each text data as bag-of-words. A high weight will be given by a high term frequency in the given document and a low document frequency of the term in the whole text corpus. TF-IDF tends to filter out common terms. For our test data sets, we first remove about 400 stop words such as "the," "of," "good," etc. Then we compute TF-IDF of the words in each document based on the given test corpus and use the TF-IDF scores as features for clustering. TF-IDF is employed as a baseline of the clustering experiments.

LDA: We use Gibbs sampling inference of LDA [Blei *et al.*, 2003] which is implemented by Mallet [McCallum, 2002] in this experiment. Two different methods are used for training the topics.

1 We train LDA and test on the same short text data. Since the two corpora are all of short texts, LDA works with extremely sparse data. We set the topic number to be the cluster number or twice the cluster number and report the better of the two. This method is denoted as "LDA #1."

2 For the news data, we also train the LDA model on long texts (the main body of the news) and test it on the

Table 2: NMI scores of the clustering experiments on news title data set.

NMI	TF-IDF	LDA #1	LDA #2	ESA	PROB. CONCEPT.	G+D CONCEPT.
COMPANY VS. DISEASE	0.303±0.017	0.176±0.159	0.300±0.121	**0.870±0.063**	0.863±0.034	0.868±0.026
COMPANY VS. ENTERTAINMENT	0.257±0.046	0.055±0.047	0.301±0.175	0.233±0.221	0.646±0.044	**0.798±0.027**
COMPANY VS. FOOD	0.224±0.091	0.077±0.074	0.323±0.024	0.712±0.065	0.636±0.053	**0.933±0.002**
COMPANY VS. POLITICIAN	0.341±0.053	0.038±0.063	0.320±0.027	0.857±0.059	0.705±0.063	**0.933±0.020**
COMPANY VS. SPORT	0.188±0.104	0.159±0.137	0.213±0.036	0.573±0.163	0.726±0.072	**0.814±0.020**
DISEASE VS. ENTERTAINMENT	0.193±0.070	0.115±0.081	0.690±0.041	**0.762±0.052**	0.681±0.037	0.729±0.076
DISEASE VS. FOOD	0.188±0.065	0.084±0.091	0.708±0.006	**0.813±0.049**	0.671±0.092	0.677±0.076
DISEASE VS. POLITICIAN	0.362±0.057	0.119±0.099	0.763±0.036	0.948±0.003	0.671±0.056	**0.951±0.010**
DISEASE VS. SPORT	0.166±0.059	0.151±0.115	0.359±0.217	**0.915±0.004**	0.747±0.057	0.888±0.011
ENTERTAINMENT VS. FOOD	0.092±0.075	0.036±0.044	0.507±0.047	0.704±0.052	0.306±0.042	**0.725±0.036**
ENTERTAINMENT VS. POLITICIAN	0.320±0.082	0.080±0.063	0.665±0.101	0.673±0.079	0.386±0.098	**0.922±0.008**
ENTERTAINMENT VS. SPORT	0.172±0.090	0.080±0.057	0.170±0.114	0.281±0.167	0.364±0.060	**0.850±0.008**
FOOD VS. POLITICIAN	0.242±0.041	0.071±0.048	0.758±0.011	0.848±0.023	0.487±0.034	**0.960±0.011**
FOOD VS. SPORT	0.227±0.057	0.078±0.065	0.213±0.106	0.810±0.006	0.454±0.100	**0.830±0.028**
POLITICIAN VS. SPORT	0.355±0.027	0.136±0.122	0.216±0.035	**0.950±0.004**	0.453±0.022	0.916±0.014
AVERAGE	0.242±0.080	0.097±0.043	0.434±0.223	0.730±0.219	0.586±0.164	**0.853±0.089**

Table 3: NMI scores of the clustering experiments on Twitter data set.

	TF-IDF	LDA #1	ESA	PROB. CONCEPT.	G+D CONCEPT.
NMI	0.468±0.057	0.267±0.057	0.522±0.018	0.568±0.067	**0.573±0.017**

short texts. We use the body field of the news articles corresponding to the titles for training. Each article has several hundreds of words. The topic number is set to be 10 or 20, and we report the better of the two. This method is denoted as "LDA #2."

ESA: We import the Wikipedia articles from the Wikipedia dump.[4] To improve ESA, we preprocess the Wikipedia articles with the following rules. First, we remove the articles less than 100 words and remove the articles less than 10 links. Then we remove all the category pages and disambiguation pages. Moreover, we move the content to the right redirection pages. Finally we obtain about one millions Wikipedia articles for indexing. We compute TF-IDF weights for word concept pairs as presented in [Gabrilovich and Markovitch, 2007]. Top 1,000, 2,000, and top 10,000 concepts are used as features for clustering, and we report the best.

Probabilistic Conceptualization (Probabilistic Concept.): We implement the method [Song *et al.*, 2011] and the top 100, 200 and 400 concepts are used for clustering respectively, and we report the best.

Generative + Descriptive Conceptualization (G+D Concept.): We compute the concept distribution $\mathbf{c}$ for each text, and use top 400 concepts in the clustering experiments.

4.2 Clustering Results

We use spherical K-means clustering on the concept (or topic) vectors generated by each method. The spherical K-means clustering results also depend on initialization (especially when the data vectors are of high dimension). In this experiment, we randomly initialize K-means and repeat clustering five times to report the result with the lowest objective function value. All the numbers reported is based on 10 random trials (each trail is based on five random initialization).

The clustering results for news title data are shown in Table 2. The normalized mutual information (NMI) [Strehl and Ghosh, 2002] scores are presented. In general, the larger the NMI scores are, the better the clustering results are. We report the results of pairwise category clustering here to check the more detailed information. From the results we can see that, LDA #1 performs worst because it is trained on very sparse short texts, where there is no enough statistical information to infer word topics. LDA #2 is better, but it still underperforms the three knowledge based methods. We can also train LDA on a very large corpus, e.g., Wikipedia, and can expect much better results. However, training LDA on very large data set is much slower than the knowledge extraction procedures used by ESA and Probase. Sometimes ESA performs best, however, it does not show significant improvement over our method. Contrarily, for the problems ESA does not perform well, i.e., "Company vs. Entertainment" and "Entertainment vs. Sport," our method works very well. Moreover, we can see that our method significantly outperforms Song *et al.*'s conceptualization method.

For the Twitter data, since we are not able to find appropriate long texts, LDA #2 is not performed. The clustering results are shown in Table 3. We can see that the results are consistent with the news title data set. Our method performs the best and shows improvement over the compared methods.

5 Conclusions

We have unified descriptive, generative, and discriminative text conceptualization in a Bayesian perspective, and discussed the advantages and problems respectively. To solve the problems, we proposed a generative + descriptive solution to short text conceptualization. The model incorporates both union and intersection operations of the concept sets for the terms detected in the short text, and results in better conceptual descriptions. We use one news title data set and one Twitter message data set to demonstrate that clustering on our conceptualization results can outperform the state-of-the-art conceptualization and topic modeling approaches.

[4] http://en.wikipedia.org/wiki/Wikipedia:Database_download

Acknowledgments

The authors would thank Professor Dan Roth and the anonymous reviewers for the helpful comments and suggestions to improve this paper. This work is partially supported by the Multimodal Information Access & Synthesis Center at UIUC, part of CCICADA, a DHS Science and Technology Center of Excellence, by the Army Research Laboratory (ARL) under agreement W911NF-09-2-0053, and by DARPA under agreement number FA8750-13-2-0008. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied by these agencies or the U.S. Government.

References

[Agrawal *et al.*, 2014] Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. Similarity search using concept graphs, rakesh agrawal. In *CIKM*, pages 719–728, 2014.

[Blei *et al.*, 2003] D.M. Blei, A.Y. Ng, and M.I. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[Chang *et al.*, 2008] Ming-Wei Chang, Lev Ratinov, Dan Roth, and Vivek Srikumar. Importance of semantic representation: Dataless classification. In *AAAI*, pages 830–835, 2008.

[Dhillon and Modha, 2001] Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42:143–175, 2001.

[Egozi *et al.*, 2011] Ofer Egozi, Shaul Markovitch, and Evgeniy Gabrilovich. Concept-based information retrieval using explicit semantic analysis. *ACM Transactions on Information Systems*, 29(2):8:1–8:34, 2011.

[Gabrilovich and Markovitch, 2006] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *AAAI*, pages 1301–1306, 2006.

[Gabrilovich and Markovitch, 2007] Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *IJCAI*, pages 1606–1611, 2007.

[Gabrilovich and Markovitch, 2009] Evgeniy Gabrilovich and Shaul Markovitch. Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research*, 34(1):443–498, 2009.

[Hua *et al.*, 2013] Wen Hua, Yangqiu Song, Haixun Wang, and Xiaofang Zhou. Identifying users' topical tasks in Web search. In *WSDM*, pages 93–102, 2013.

[Huang *et al.*, 2012] Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. Improving word representations via global context and multiple word prototypes. In *ACL*, pages 873–882, 2012.

[Kim *et al.*, 2013] Dongwoo Kim, Haixun Wang, and Alice Oh. Context-dependent conceptualization. In *IJCAI*, pages 2654–2661, 2013.

[Koller and Friedman, 2009] D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[Li *et al.*, 2013] Pei-Pei Li, Haixun Wang, Kenny Q. Zhu, Zhongyuan Wang, and Xindong Wu. Computing term similarity by large probabilistic isa knowledge. In *CIKM*, pages 1401–1410, 2013.

[Liu *et al.*, 2012] Xueqing Liu, Yangqiu Song, Shixia Liu, and Haixun Wang. Automatic taxonomy construction from keywords. In *KDD*, pages 1433–1441, 2012.

[McCallum, 2002] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[Minka, 2003] T.P. Minka. Estimating a Dirichlet distribution. *Annals of Physics*, 2000(8):1–13, 2003.

[Salton and McGill, 1983] G. Salton and M.J. McGill. *An Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[Song and Roth, 2014] Yangqiu Song and Dan Roth. On dataless hierarchical text classification. In *AAAI*, pages 1579–1585, 2014.

[Song and Roth, 2015] Y. Song and D. Roth. Unsupervised sparse vector densification for short text similarity. In *NAACL*, 5 2015.

[Song *et al.*, 2011] Yangqiu Song, Haixun Wang, Zhongyuan Wang, Hongsong Li, and Weizhu Chen. Short text conceptualization using a probabilistic knowledgebase. In *IJCAI*, pages 2330–2336, 2011.

[Song *et al.*, 2014] Yangqiu Song, Haixun Wang, Weizhu Chen, and Shusen Wang. Transfer understanding from head queries to tail queries. In *CIKM*, pages 1299–1308, 2014.

[Strehl and Ghosh, 2002] Alexander Strehl and Joydeep Ghosh. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal on Machine Learning Research*, 3:583–617, 2002.

[Wang *et al.*, 2014] Zhongyuan Wang, Fang Wang, Ji-Rong Wen, and Zhoujun Li. Concept-based short text classification and ranking. In *CIKM*, pages 1069–1078, 2014.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q. Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492, 2012.

[Zhu, 2003] Song Chun Zhu. Statistical modeling and conceptualization of visual patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):691–712, 2003.