Making Sense of Unstructured Text Data

Article · April 2017

William Campbell
Amazon

Making #Sense of #Unstructured Text Data*

Lin Li, William M. Campbell, Cagri Dagli, Joseph P. Campbell
MIT Lincoln Laboratory
lin.li@ll.mit.edu, wcampbell@ll.mit.edu

Abstract

Many network analysis tasks in social sciences rely on pre-existing data sources that were created with explicit relations or interactions between entities under consideration. Examples include email logs, friends and followers networks on social media, communication networks, etc. In these data, it is relatively easy to identify who is connected to whom and how they are connected. However, most of the data that we encounter on a daily basis are unstructured free-text data, e.g., forums, online marketplaces, etc. It is considerably more difficult to extract network data from unstructured text. In this work, we present an end-to-end system for analyzing unstructured text data and transforming the data into structured graphs that are directly applicable to a downstream application. Specifically, we look at social media data and attempt to predict the most indicative words from users' posts. The resulting keywords can be used to construct a context+content network for downstream processing such as graph-based analysis and learning. With that goal in mind, we apply our methods to the application of cross-domain entity resolution. The performance of the resulting system with automatic keywords shows improvement over the system with user-annotated hashtags.

1 Introduction

Automatic extraction of intelligent and useful information from data is one of the main goals in data science. Traditional approaches have focused on learning from structured features, i.e., information in a relational database. However, most of the data encountered in practice are unstructured (i.e., forums, emails and web logs); they do not have a predefined schema or format. The challenge of learning and extracting relevant pieces of information from unstructured text data is an important problem in natural language processing and can render useful graph representation for various machine learning tasks.

Various efforts have been proposed to develop algorithms for processing unstructured text data. At a top level, text can be either summarized by document level features (i.e., language, topic, genre, etc.) or analyzed at a word or sub-word level. Text analytics can be unsupervised, semi-supervised, or supervised.

In this work, we focus on word analysis and examine *unsupervised* methods for processing unstructured text data, extracting relevant information, and transforming it into structured relations that can then be leveraged in graph analysis and various other downstream applications. Unsupervised methods require less human annotation and can easily fulfill the role of automatic analysis. The specific application that we examine in this work is the problem of associating entities (typically people and organization) across multiple platforms, also known as the *cross-domain entity resolution* [1].

Consider social media platforms as an example. Information about an entity is often present in multiple forms, such as profile, social activities and content. Profiles give information about the username, full name, profile pictures, etc. Social activities (i.e., mentions, comments, re-posts, etc.) provide information about the interaction among entities on social media. Content covers almost everything a user posts on social media. One prominent example is the use of hashtags. Hashtags can be considered as user-annotated keywords that are indicative of the content or the topic of the posts. Therefore, hashtags are often introduced in the graph construction process that mix together content features and context features. The resulting context+content graph is often able to provide better and more interpretable results. Yet hashtags are generally noisy because there is no restriction on how they can be used in a post [2]. For example, users may mark the word `boston` as a hashtag (i.e., `#boston`) in one post, but not use it as a hashtag in other posts. Additionally, some users tend to use too many hashtags in a post while others may not use hashtags at all. In the case of forums, emails, web logs or online marketplaces, there simply exists no user-annotated hashtags.

*This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

The goal of this work is to investigate methods for learning a graph representation of noisy and unstructured text data via automatic hashtag annotation and contrast it to standard methods which are based on user-annotated hashtags. Automatic hashtag annotation is a way of consistently extracting indicative content words from the posts. These words are then used to construct a graph relation based on user mentions, co-occurrences and user interactions. We refer to the resulting graph as a content+context graph. Automatic hashtag annotation is closely related to automatic tag extraction [3, 4] and keyword extraction [5, 6]. Although here we focus on the problem of structuring social media data, the methods for automatic hashtag annotation and representing the data in graph form are more general and also applicable to non-social media platforms, such as forums, emails, and marketplaces.

2 Structuring Unstructured Text Data

The proposed framework for structuring unstructured text data is shown in Figure 1. It differs from previous methods by offering an end-to-end process of analyzing unstructured text data via automatic hashtag annotation, building graphs of entities and hashtags, and performing graph analyses that are directly applicable to downstream applications. For text analysis, we focus on methods for finding relevant words in the text. Specifically, we look at social media data (Twitter and Instagram) and attempt to predict indicative content words for users' posts. The resulting words are then used for constructing graphs that are relevant to the downstream application. The specific application that we examine in this work is the problem of associating entities (typically people and organization) across multiple social media sites, known as cross-domain entity resolution; see Section 3.2.

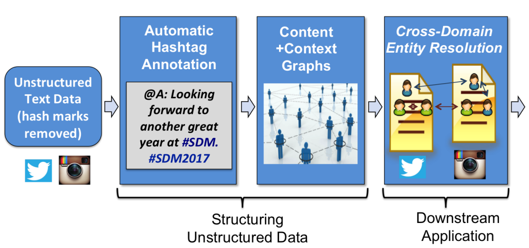


Figure 1: Proposed framework for unstructured text analysis

2.1 Data Sets Twitter and Instagram data from the Boston area were collected for our experiments [7, 1]. Twitter data consist of approximately 4.65 million tweets collected from 1/1/2014 to 10/1/2014. Instagram data consist of 3.71 million posts (and comments) collected between 12/31/2013 to 12/31/2014. For Instagram, some comments on these posts extend into 2015.

2.2 Pre-processing Our pipeline for pre-processing the social media content data is as follows. First, we perform text normalization on the string; it converts any unusual UTF-8 characters to a standardized form, eliminates emojis, emoticons, links and retweets. Then we apply standard text pre-processing steps: tokenization, punctuation and stop word removal. Lastly, we remove all the hash marks from the string. Hence, what used to be a hashtag is now a regular word in the post*.

2.3 Automatic Hashtag Annotation For hashtag annotations, we explore two different but related strategies: a topic-based method and a community-based graph method.

Our pipeline for the topic-based method is as follows. First, we extract word counts from users' posts and perform PLSA topic analysis [8] at the topic level. Then given a latent topic variable c, each word w is associated with a probability $p(w|c)$. Second, we extract the most relevant words from each topic, measured by the probability $p(w|c)$. Each of these words is then annotated as a hashtag in the posts. This approach differs from the TF-IDF [9] and other word-frequency based methods for word importance. It enables a deeper coverage across a range of topic areas in the posts. For example, in addition to marking top-ranked words belonging to a large or more popular topic as hashtags, we also include words that are highly relevant to a small or less popular topic cluster. Table 1 shows an example list of automatically annotated hashtags using the Twitter dataset described in Section 2.1.

Table 1: Top-ranked words from selected topic using the topic-based method

Topic	Interpretation	Automatic Hashtags
1	Jobs	job, tweetmyjobs, greater, ymca, part, education, boston, group
2	Food	lunch, eat, breakfast, sweet, wine, cafe
3	World Cup	world, usa, top, worldcup, worldcup2014, cup, goal, final, beat, track

Another related approach for automatic hashtag annotation is the community-based graph method. This methods attempts to find clusters of words via word co-occurrence relationships. First, we construct a weighted

*Although a hashtag may contain multiple words, in this work, we treat it as a phrase with no space separating the words.

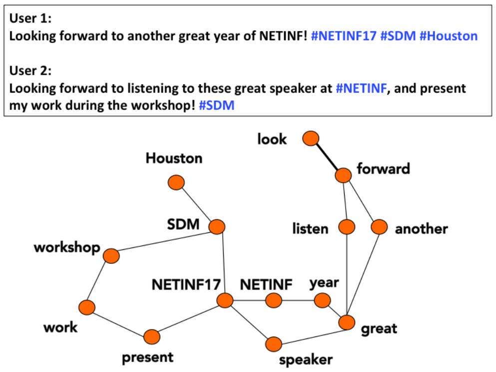


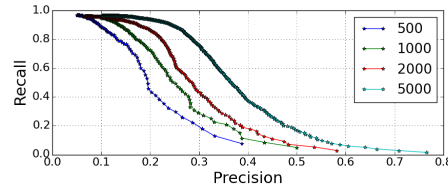
Figure 2: Word co-occurrence graph construction

word co-occurrence graph, where vertices in the graph represent words and edges represent co-occurrence of words in users' posts; see Figure 2. We then perform community detection on the co-occurrence graph using infomap [10]. From each community, we extract words with the highest PageRank values and then annotate them as hashtags.

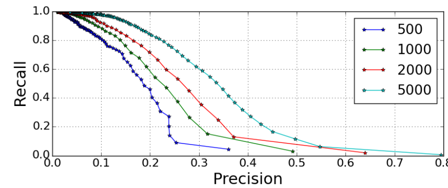
2.4 Content+Context Graph Construction Graph construction is performed by designating both users and the automatically annotated hashtags as vertices and representing edges as four types of interactions: user-to-user posts, user mentions of hashtags, reposts and co-occurrence of users or hashtags. The count of occurrence of different edge types is saved in the graph. For analysis, counts are summed across edge types. The resulting content + context graph is weighted and undirected. It has been shown to be useful in a number of applications. In the experiment, we construct two graphs, G_{twitter} for the Twitter graph and G_{inst} for the Instagram graph, and showcase its usefulness through an important application, cross-domain entity resolution [1].

3 Experiments

3.1 Automatic Hashtags We perform experiments on Twitter and Instagram corpora as described in Section 2.1. We follow the process presented in Section 2.2 for pre-processing to clean up the data and remove user-annotated hashtags. Then, we use both the topic-based and community-based methods for automatic hashtag annotation. We show how the automatic hashtags compare with the user-annotated hashtags. Fig. 3 presents precision and recall of recovering the hashtag vocabulary. Each curve in Fig. 3 is generated for the top-M most common user-annotated hashtags where $M = 500$, 1000, 2000, and 5000. True positives are defined as automatically selected hashtags that are in the top-M user-annotated hashtags. If tp is the number of true



(a) Topic-based method



(b) Community-based method

Figure 3: Precision and recall curves of the automatic hashtags against top M user-annotated hashtags.

positives and K is the number of automatically selected hashtags, then precision is tp/K and recall is tp/M.

The number of automatically selected hashtags, K, can easily be varied with both the graph and topic methods by varying the number of words selected per cluster. For small K, precision is high, but recall is low since less common user-annotated hashtags are missed. For large K, our methods recall all of the top-M user-annotated hashtags. Note that for a fixed recall, as M increases, precision also increases. This property indicates that the automatic methods are finding user-hashtags, but not in the same order as the top-M most common user-annotated ranking.

3.2 Cross-Domain Entity Resolution System We define cross-domain entity resolution as the problem of matching entities across different platforms. Prior approaches to the entity resolution problem include [11, 12, 13, 14, 15, 1]. We use the framework studied in [1] to extract features for associating entities across Twitter and Instagram. Two classes of features are extracted: username similarity and graph-based features. For username similarity, we use the standard Jaro-Winkler similarity [16, 17] for approximating username matches. It has been shown in [1] to be best suited for username matching.

Graph-based features include community membership match and weighted neighborhood match between pairs of Twitter and Instagram accounts. The pipeline for computing the community membership match is as follows. First, we use the automatically-extracted hashtags and users' interactions to construct both a Twitter graph G_{twitter} and an Instagram graph G_{inst}; See Section 2.4. Second, we use the technique described

Table 2: Summary of entity resolution results. 'P' denotes the profile feature, 'N' denotes the content + context graph feature, and 'P+N' denotes the fusion of the two. The interpolated value is given by *.

Fusion	**Hashtags**	**Method**	**EER ALL (%)**	**EER NT (%)**
P			1.54	5.74*
P+N	User-annotated Hashtags		1.16	3.79
P+N	4K Automatic Hashtags	Topic	**1.17**	3.48
P+N	4K Automatic Hashtags	Community	1.19	**3.39**
P+N	10K Automatic Hashtags	Topic	1.17	3.51
P+N	10K Automatic Hashtags	Community	**1.1**	**3.24**

in [7, 1] to merge the two graphs via common hashtags. The resulting graph is denoted G_{merged}; it includes all the nodes from both G_{twitter} and G_{inst}. Third, we apply community-based detection on the merged graph G_{merged}. Lastly, for each pair of users across Twitter and Instagram, we compute their graph features: community similarity score and weighted neighborhood similarity score. See Appendix A for more details.

The graph features are then combined with the username match to obtain a fused entity resolution system. We use a random forest to train a fuser with inputs being the vector of the three similarity scores for each pair users and the output being the probability of a match. For evaluation, we construct two sets of trials for training and testing our system. Each trial consists of a user pairs (U_t, U_i) where U_t denotes a user from Twitter and U_i denotes a user from Instagram. A trial with a label '1' implies that U_t and U_i represent the same real-world entity, while a trial with a label '0' means they are different entities. The performance of cross-domain entity resolution system is measured by the standard miss and false alarm rates. The resulting equal error rates (EER) for all trials and non-trivial trials (NT) are shown in Table 2. Non-trivial trials are trials where the usernames are not an exact string match. Observe that systems using automatic hashtag analysis perform significantly better than the profile-only system. Also, graph construction using automatic hashtags improves performance over a system using graphs constructed from user-annotated hashtags.

4 Conclusion

We presented an end-to-end system for analyzing unstructured text data and transforming the data into structured graphs that can be directly applied to various downstream applications. For text analysis, we presented two unsupervised methods for automatic hashtag annotation: a topic-based method and a community-based graph method. Both methods show promising results in predicting relevant words from text. We also build content + context graphs using the automatically-annotated hashtags and use them for cross-domain Twitter/Instagram entity resolution. The performance of the resulting system outperforms the system using the original user-annotated hashtags.

References

[1] WM Campbell, Lin Li, C Dagli, J Acevedo-Aviles, K Geyer, JP Campbell, and C Priebe. Cross-domain entity resolution in social media. In *The 4th International Workshop on Natural Language Processing for Social Media*, 2016.

[2] Dmitry Davidov, Oren Tsur, and Ari Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *Proc of the 23rd int conf on comput linguistics*, pages 241–249. Ass for Comput Linguistics, 2010.

[3] R. Krestel, P. Fankhauser, and W. Nejdl. Latent dirichlet allocation for tag recommendation. In *Proc. of the 3rd ACM conf. on Recommender sys.*, pages 61–68. ACM, 2009.

[4] Z. Liu, X. Chen, and M. Sun. A simple word trigger method for social tag suggestion. In *Proc. of the Conf. on Empirical Met. in Nat. Lang. Proc.*, pages 1577–1588. ACL, 2011.

[5] R. Mihalcea and P. Tarau. Textrank: Bringing order into texts. In *Proc of EMNLP*, pages 404–411. ACL, 2004.

[6] W. X. Zhao, J. Jiang, J. He, Y. Song, P. Achananuparp, E.-P. Lim, and X. Li. Topical keyphrase extraction from twitter. In *Proc. of the 49th An. Meet. of ACL: Human Lang. Tech.-Vol. 1*, pages 379–388. ACL, 2011.

[7] Lin Li and William M. Campbell. Matching community structure across online social networks. In *Proceedings of the NIPS Workshop: Networks in the Social and Information Sciences*, 2015.

[8] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proc. of 15th conf on Uncertainty in artificial intel*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999.

[9] Gobinda Chowdhury. *Introduction to modern information retrieval.* Facet publishing, 2010.

[10] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical review E*, 80(5):056117, 2009.

[11] Lise Getoor and Ashwin Machanavajjhala. Entity resolution in big data. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[12] Danai Koutra, Hanghang Tong, and David Lubensky. Big-align: Fast bipartite graph alignment. In *Data Mining (ICDM), IEEE 13th Int Conf on*, pages 389–398, 2013.

[13] Tereza Iofciu, Peter Fankhauser, Fabian Abel, and Kerstin Bischoff. Identifying users across social tagging systems. In *ICWSM*, 2011.

[14] Vince Lyzinski, Daniel L Sussman, Donniell E Fishkind, Henry Pao, Li Chen, Joshua T Vogelstein, Youngser Park, and Carey E Priebe. Spectral clustering for divide-and-conquer graph matching. *Parallel Comput*, 47:70–87, 2015.

[15] Olga Peled, Michael Fire, Lior Rokach, and Yuval Elovici. Entity matching in online social networks. In *SocialCom, Int Conf on*, pages 339–344. IEEE, 2013.

[16] Matthew A Jaro. Probabilistic linkage of large public health data files. *Stats in med*, 14(5-7):491–498, 1995.

[17] William E Winkler. The state of record linkage and current research problems. In *Stat Res Div, US Census Bureau*. Citeseer, 1999.

[18] Martin Rosvall and Carl T Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc of Nat Acad of Sci*, 105(4):1118–1123, 2008.

[19] David Liben-Nowell and Jon Kleinberg. The link prediction problem for social networks. In *Proc of 12th CIKM*, pages 556–559. ACM, 2003.

A Appendix

A.1 Cross-Domain Community Detection We have constructed a Twitter graph G_{twitter} and an Instagram graph G_{inst}. To extract community features, we perform cross-media community detection to identify communities simultaneously across Twitter and Instagram graphs. The key to achieving this is to align the graphs using *seeds*. Seeds are known vertex matches across graphs (e.g., the common hashtags). We use a random walk-based approach to align the graphs to form a single interconnected graph. There are three general strategies: (1) *aggregation* that merges pairs of vertices in the seed set; (2) *linking* that adds links to the seed pairs; and (3) *relaxed random walk* that allows a random walker to switch between graphs with some probability. Once the graphs are aligned and connected, it is straightforward to adapt Infomap [18] for community detection. Infomap is a random walk-based algorithm that partitions the graph by minimizing an information-theoretic based objective function.

For the experiment, we use the aggregation approach with Infomap for community detection across Twitter and Instagram graphs. Prior work [7] shows that with a sufficient number of seeds, the aggregation approach is the most faithful to the underlying community structure. Specifically, we first associate a Markov transition matrix to the *union* of G_{twitter} and G_{inst}. Each element in the Markov matrix represents the probability of a random walk of length 1 going from one vertex to the other; the Markov transition probability is computed by normalizing the edge weights between a vertex and all of its adjacent vertices. Second, for each vertex pair in seeds, we merge the two vertices and update the transition matrix with probability $p = 0.5$ that a random walk moves to the adjacent vertices in G_{twitter} and probability $1-p$ that a random walk moves to the adjacent vertices in G_{inst}. The resulting aligned and connected graph is denoted as G_{join}; it includes all the vertices from both G_{twitter} and G_{inst} and the edge weights are given by the Markov transition matrix. Additionally, we apply Infomap only on the largest connected component of the aligned graph G_{join}; vertices that are in the largest connected component have a community assignment.

A.2 Graph-based Features As hinted earlier, we are interested in extracting two classes of graph features: community features and neighborhood features. Note that neighbors of a vertex in a graph are vertices connected by an edge to the specified vertex; they are also referred to as 1-hop neighbors. Generally, for a vertex v in a graph, k-hop neighbors are defined as vertices that are reachable from v in *exactly* k hops.

A.2.1 Community Features and Similarity The basic idea is to be able to represent the similarity in community membership between users across graphs. First, we perform a cross-media community detection on Twitter and Instagram graphs. One simple way to compare community features of two users is to assign a value '1' to the pair that are in the same community and '0' otherwise. However, this binary-valued similarity score will likely cause confusion because it assigns a similarity score '1' to all users belonging to the same community. To mitigate this problem, we propose to represent a user's community feature via the community membership of all its (k-hop) neighbors in its respective graph. For example, the community feature for a Twitter user U is given by a count vector $\mathbf{c}(U)$ with entries

$$c_i = |\{N|\,\text{comm}(N) = i,\ N \in \text{nbr}(U|G_{\text{twitter}})\}| \tag{A.1}$$

where comm$(\cdot)$ indicates the community assignment of vertex N and nbr$(\cdot)$ is the set of k-hop neighbors.

For the experiment, we set $k = 1, 2$ and use one of the two methods to measure the similarity in community feature between users. One is to compute the dot product of normalized count vectors, i.e., $\text{sim}(\mathbf{c}(U_i), \mathbf{c}(U_j)) = \frac{\mathbf{c}(U_i)^T \mathbf{c}(U_j)}{\|\mathbf{c}(U_i)\|_2 \|\mathbf{c}(U_j)\|_2}$.

A.2.2 Neighborhood Features and Similarity Given the aligned and connected graph G_{join} (i.e., constructed from joining G_{twitter} and G_{inst} using seeds), we seek to compute the similarity between two users by analyzing the proximity of their corresponding vertices in G_{join}. A popular approach is based on vertex neighborhoods [19], e.g., common-neighbors approach and its variants. The approach here is similar. However, instead of simply counting the number of common neighbors, we represent the neighborhood feature using the transition probability of the random walk in G_{join}. Specifically, the neighborhood feature of a Twitter user U is given by $\mathbf{p}(U)$, whose i^{th} entry $p_i = p(U, U_i)$ represents the probability that a random walk of a specified length k starting at U ends at the i^{th} vertex U_i in G_{join}. The neighborhood similarity is given by the normalized dot product of the neighborhood features.

We choose to use $k = 1$ hop for computing the neighborhood similarity. Note that for $k = 1$, the edge probability $p(U_1, U_2) = 0$ if U_1 and U_2 are not connected. Also, isolated vertices are not considered.