Learning Temporal Tagging Behaviour

Toni Gruetze, Gary Yao, and Ralf Krestel
Hasso Plattner Institute
Potsdam, Germany
{firstname.lastname}@hpi.de

ABSTRACT

Social networking services, such as Facebook, Google+, and Twitter are commonly used to share relevant Web documents with a peer group. By sharing a document with her peers, a user recommends the content for others and annotates it with a short description text. This short description yield many chances for text summarization and categorization. Because today's social networking platforms are real-time media, the sharing behaviour is subject to many temporal effects, i.e., current events, breaking news, and trending topics. In this paper, we focus on time-dependent hashtag usage of the Twitter community to annotate shared Web-text documents. We introduce a framework for time-dependent hashtag recommendation models and introduce two content-based models. Finally, we evaluate the introduced models with respect to recommendation quality based on a Twitter-dataset consisting of links to Web documents that were aligned with hashtags.

1. INTRODUCTION

In recent years, collaborative tagging in Web 2.0 services such as Delicious (http://del.icio.us) has emerged as an efficient way to organize large collections of documents. The principle idea is that documents are collectively labelled with freely chosen categories (tags) by users. A plethora of research has been done to automatically recommend appropriate tags for arbitrary documents [3]. These recommendations support the user and facilitate the organization of documents by applying more concise tags. However, tags found in such collaborative tagging systems are inherently diverse. This makes automatic tag recommendation a challenging task.

On Twitter, one of the most frequented microblogging service, hashtags have emerged as a means of classifying shared content. Because hashtags are hyperlinked to search results of equally annotated tweets, they are an important means for grouping tweets according to topics. In this paper we investigate hashtag recommendation for tweets that contain URLs. This holds various challenges: First, we need to identify suitable hashtags given a URL. That means we need to learn the context of a hashtag. Secondly, we need to account for a context shift of a hashtag. And finally, we need to ensure our approach scales well.

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2741701.

Context Learning. An hashtag recommendation approach based on the linked contents (i.e., URLs to Web pages) on Twitter can be compared with "traditional" tag recommendation approaches. However, in contrast to the tag usage in social bookmarking systems such as Delicious, the usage of hashtags in Twitter has proven to be guided by current events. The meaning behind all hashtags has to be learned, that is in which context was a hashtag used before. This ensures that we can recommend relevant hashtags given a new URL. For instance, as of February 2015, the hashtag #SuperBowl should be strongly related to articles about the American football teams of the New England Patriots and the Seattle Seahawks. A hashtag might be ambiguous and thus cover different topics, e.g. #football could be used in the context of the Super Bowl but also in the context of European soccer.

Context Drift. The temporal aspect of users' tagging behaviour is very important in a highly dynamic and volatile system such as Twitter. The contexts, in which certain hashtags are used, changes often very fast and a context drift is the result[1]. Therefore, building an recommendation model that inherently incorporate temporal changes of the meanings of hashtags is essential. The model has to adopt to temporal changes of the hashtag meanings. For instance, the hashtag #MissyElliott received a big context drift towards American football during Super Bowl XLIX. A hashtag recommender has to quickly adopt to these changes and might recommend this hashtag for further articles about the Super Bowl.

Scalability. The model has to be capable of tracking changes in Twitter community behaviour. Hence, it has to quickly update the meanings and contexts of hashtags to enable appropriate recommendations. In the case of popular events, hundreds of URLs per second are shared on Twitter which shows that scalability plays an important role for recommendation models based on highly dynamic platforms.

Contribution. The focus of this work is on the development of scalable hashtag recommendation models that aim at adopting to currents developments and trending topics in the Twitter community. We explicitly model context drift by incorporating temporal information. As discussed by Kwak et al. Twitter trends differ from the ones stemming from other mass media, such as CNN [8].

In the next section, we discuss related work. In Section 3 we introduce our time-adapting hashtag recommendation followed by the results of our evaluation in Section 4.

[1]Hu et al. provide an extensive study on Twitter traffic from May 1st, 2011 after the first rumours of the death of Osama Bin Laden [5]. The study consists of approximately 600k tweets containing the term "laden", shared within the first two hours after the first rumour. It is obvious, that the contents shared with the hashtags #BinLaden or #Obama were subject to an enormous topic drift that day.

2. RELATED WORK

Previous work on Twitter hashtag recommendation is mainly based on the tweet text and aims at recommendation for arbitrary tweets. Tweet text inherently differs from Web-text documents, such as news or blog articles, due to the 140-character limit per tweet, which frequently results in usage of abbreviations, phonetic substitutions, and emoticons [4]. Moreover, the fact that topics on Twitter are under constant change is not dealt with in previous work on hashtag recommendation.

Mazzia and Juett [11] suggest to recommend hashtags using a Naive Bayes model based on a binomial distribution over the top 50,000 words occurring in the tweet texts. Given a tweet represented as binary features (term occurrence), the most appropriate hashtag is derived by a maximum likelihood estimate over these features.

A hashtag recommendation model based on similarities between tweet texts has been proposed by Zangerle et al. [14]. Hashtags recommended for a tweet are computed by firstly retrieving the most similar tweets. The hashtags belonging to the retrieved tweets are the recommendation candidates. Secondly, the hashtags are ranked by different scoring functions. Kywe et al. [9] extended this approach by including recently used hashtags of users with similar preferences, where user preferences are modelled based on the usage frequency of hashtags.

In contrast to the mentioned approaches, this work focuses on recommending hashtags for documents linked-to in tweets, such as newspaper articles, blog entries, etc. Hence, we treat tweets as comments for actual shared documents. Sedhai and Sun follow a similar goal [12]. Their work compares different hashtag recommendation strategies for *hyperlinked tweets*, where the features are based on the tweet texts, the linked text contents, as well as the named entities[2] found in the linked documents. In contrast to their work, we additionally focus on the context drift of each hashtag, which comes from temporal dynamics in the tagging behaviour of the Twitter community. We explicitly design a recommendation models being capable of adapting to topical changes.

Apart from hashtag recommendation there is a plethora of work on tag recommendation in collaborative tagging systems [2]. Social bookmarking systems, such as Delicious, enable users to add tags to resources (e.g., Web pages). Recommending tags in this setting is highly comparable to the task of hashtag recommendation for tweets containing URLs. But note that tags in a collaborative environment often have different requirements compared to hashtags. While tagging systems are often used to organize bookmarks, e.g. categorize a Web page into category 'politics', hashtags are used in a much more specific way, e.g. #ObamaCare, or to express an opinion, e.g. #DalaiLamaStopLying. An overview of recommendation algorithms for social bookmarking systems can be found in [13]. These range from content-based approaches [1], over collaborative filtering approaches [6], to tag co-occurrence approaches [7]. Temporal aspects are typically not taken into account in tag recommendation for collaborative tagging systems. Furthermore, recommending hashtags is strongly influenced by the popularity of hashtags. Hence, hashtag recommendation has to consider the community preferences instead of strictly focusing on the documents content. For instance, news reports about the NFL game of the Indianapolis Colts against the New England Patriots might be tagged by #football, #Patriots, #Colts, #NFL, ..., whereas the most appropriate hashtag for the Twitter community might be #Deflategate.

[2]based on the Stanford NER tagger

3. TEMPORAL TAGGING BEHAVIOUR

Next, we discuss an abstract hashtag recommendation model, capable of covering temporal developments such as context drifts. Given a tweet $\mathcal{T}$ shared at time $t_\mathcal{T}$, covering a set of hashtags $H_\mathcal{T}$, and a set of linked documents (i.e., hyperlinks from the tweet text to external text contents) $D_\mathcal{T}$, we define the set of alignments stemming from $\mathcal{T}$ as

$$A_\mathcal{T} = \{t_\mathcal{T}\} \times H_\mathcal{T} \times D_\mathcal{T}.$$

Hence, each alignment $a_x \in A_\mathcal{T}$ is a triple of sharing time, hashtag, and URL $(t_{a_x}, h_{a_x}, d_{a_x}) : t_{a_x} = t_\mathcal{T}, h_{a_x} \in H_\mathcal{T}, d_{a_x} \in D_\mathcal{T}$. For the following considerations, we ignore the dependencies between the alignments $A_\mathcal{T}$ of tweet $\mathcal{T}$. We refer to a set of arbitrary alignments (i.e., stemming from various tweets) as $\mathcal{A}$ and the set of existing hashtags as $\mathcal{H}$.

Next, we define an abstract recommendation model $\mathcal{M}$. Each model has to be capable of recommending k hashtags for a given document d:

$$\mathit{rec}_k(d, \mathcal{M}) \mapsto (h_1, \ldots, h_k); h_{1..k} \in \mathcal{H}.$$

As mentioned earlier, each model has to be capable of tracking current developments in social network platforms. Therefore, has to be extendable:

$$\mathit{add}(a, \mathcal{M}) \mapsto \mathcal{M}'.$$

Each call of *add* will update $\mathcal{M}$, such that the new information contained in a are considered for the subsequent recommendations of $\mathcal{M}'$. Following this definition, the initial training of an uninitialized model $\mathcal{M}_\varnothing$ with a set of alignments $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$ can be written as a composition:

$$\mathit{add}(a_n, \ldots \mathit{add}(a_2, \mathit{add}(a_1, \mathcal{M}_\varnothing)) \ldots) \mapsto \mathcal{M}.$$

For brevity, we will refer to such a function composition to extend $\mathcal{M}_\varnothing$ by alignment set $\mathcal{A}$ as: $\mathcal{M} = \mathit{add}_\mathcal{A}(\mathcal{M}_\varnothing)$.

Depending on the actual model, the continuous extension by more and more alignments leads to two problems: *(i)* The model covers many outdated alignments, where the hashtag context grows unlimited and gets noisy. For instance, the visit of Barack Obama in India end of January 2015 changed the context of #Obama towards New Delhi. However, recommendations for documents about New Delhi might ignore this correlation one month later. *(ii)* The resource consumption of the model will grow with respect to the contained alignments, making the recommendation process more complicated. Hence, a recommendation model has to be capable of forgetting alignments. Analogously to *add*, we define the function $\mathit{rem}(a, \mathcal{M})$, that updates $\mathcal{M}$, such that the information covered by a is removed from the model.

3.1 Update strategies

So far, we defined the requirements for an abstract recommendation model capable of recommending hashtags for a given document (rec_k) and tracking changes in meanings of hashtags by adding and removing new alignments (*add* / *rem*). Next, we introduce two different update strategies, that enable the model to track temporal developments and context drifts of hashtags.

Given a data stream of alignment items (following alignment stream) $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \ldots)$ arriving in a natural, chronological order ($t_{\mathbf{a}_1} \leq t_{\mathbf{a}_2} \leq t_{\mathbf{a}_3} \ldots$) from a source such as Twitter. In theory, assuming a potential infinite length of $\mathbf{A}$, which is based on the hypothesis that future generations will keep using Twitter as a news sharing platform, it is obvious that neither storing $\mathbf{A}^\infty$ nor building the recommendation model $\mathcal{M} = \mathit{add}_{\mathbf{A}^\infty}(\mathcal{M}_\varnothing)$ is possible. For practical reasons, even a limited alignment stream $\mathbf{A}$ will lead to

computational difficulties. Based on English tweets collected via the Twitter Streaming API[3], we estimate number alignments per seconds to 200 on average. This sums up to a total of over 6 billion alignments per year. Considering, that our main interest lies in the textual contents and knowing that the average Web page contains 60 kilobytes of bare HTML content[4], this leads to a pure shared text content size of approximately 0.4 petabytes per year. This is a data size difficult to process with current hard- and software, whereas the question arises whether old contents are still relevant. Hence, we propose two basic update strategies:

Fixed time. This strategy is based on the assumption, that the relevance of all alignments decreases evenly. Given a predefined time threshold θ_t, the stream of relevant alignments is limited by the current time t_0 and $t_0 - \theta_t$. Thus, given an arbitrary stream $\mathbf{A}$, we define the set of relevant alignments as:

$$\mathcal{A} = \left\{ \mathbf{a}_i \in \mathbf{A} : t_0 - \theta_t \leq t_{\mathbf{a}_i} \leq t_0 \right\}$$

Assuming a uniform distribution of the alignments over time, this model is easily transformable into a queued First In First Out approach, where the size of the queue is defined by the the number of alignments within the relevance time span θ_t (or simply $\overline{|A|}$). However, the semantic of a maximal relevance time span defined by θ^t is more coherent. In our experiments, we set θ^t to a value of 21 days. This setting yielded balanced recommendation quality for all analysed models.

Hashtag queue. Different hashtags underlie different types of social attention [10]. For instance, *peak* hashtags like #TGIF (Thank God it's Friday) or #Friday13th) are only used within a small time frame, i.e., on Fridays or just even between one and three days a year. In contrast, constant used hashtags like #Obama are more uniformly used. Due to this observations, this strategy follows the assumption that it is necessary to sample a specific amount of alignments per hashtag to be able to provide good recommendations of hashtag h'.

$$\mathcal{A}_{h'} = \left\{ \mathbf{a}_i : h_{\mathbf{a}_i} = h', \left| \left\{ \mathbf{a}_j : h_{\mathbf{a}_j} = h', j > i \right\} \right| < \theta_h \right\}$$

For brevity, we omitted $\mathbf{a}_i, \mathbf{a}_j \in \mathbf{A}$ and $t_{\mathbf{a}_i}, t_{\mathbf{a}_j} \leq t_0$ in the previous equation. In our experiments, we determined a value of 130 alignments per hashtag for θ^h.

3.2 Recommendation models

Next, we provide two different recommendation strategies fulfilling the previously introduced requirements for a recommendation model.

HSD. The basic intuition behind the first model (Hashtags of Similar Documents) is that documents with similar contents are tagged similarly. Hence, given a query document d', a similar document d_{a_i} contained in the relevant alignments ($a_i \in \mathcal{A}$) covers similar concepts and topics as d'. The model is inspired by the approach of Zangerle et al. [14]. To determine the similarity, between two documents, we use the cosine similarity over tf-idf weighted term vectors of the documents. The approach extracts the top-θ_r relevant alignments A_r (based on the aligned documents) and then weights the aligned hashtags as follows:

$$\mathit{weight}(h, d') = \sum_{a \in A_r : h_a = h} \cos(d_a, d')$$

In our experiments, a value of $\theta_r = 120$ yielded good results. Following, the recommendation result is defined as:

$$\mathit{rec}_k(d', \mathcal{M}_{\mathit{HSD}}) = \underset{\{h_1..h_k\} \subset \mathcal{H}}{\arg\max} \sum_{i=1}^{k} \mathit{score}(h_i, d')$$

Implementation-wise, we decided to store the relevant documents of the model in an inverted index (i.e., Lucene[5]). To improve the retrieval time of the top-θ_r similar alignments to a document d', we restricted the query vector $q_{d'}$ to the most important terms from d'. The importance is measured in terms of tf-idf value (the length of d'_q was empirically set to 25). The top-θ_r documents were then retrieved from the index based on $q_{q'}$, and weighted based on the cosine similarity to d'. To update the set of relevant alignments $\mathcal{A}$, the respective documents as well as the aligned hashtags where *add*ed to (or *rem*oved from) the index.

The *scalability* of the model can be to achieved by applying available distributed inverted index solutions, such as Apache Solr or Elasticsearch[6]. These solutions provide a horizontal scaling with respect to concurrent write operations and dataset size by sharding the index and distributing the shards across multiple cluster nodes. Furthermore, the number of concurrent read operations can be increased by introducing replicas of the shards on further cluster nodes.

ALM. The second model (Array of Language Models) is a probabilistic approach and is based on the intuition, that each hashtag h' models a specific word distribution describing its actual meaning. To efficiently exploit the textual information, this model builds on statistical language models. Language models have been widely used in information retrieval for ranking result documents to keyword queries [15]. For this work we employ unigram language models. However, the idea might be applied using other types of language models (e.g., bigrams). Writing an article related to hashtag h' is considered to be a sampling of contents (i.e., terms) from the language model. Hence, a given alignment $\mathbf{a}_i$ with $h_{\mathbf{a}_i} = h'$ the aligned document $d_{\mathbf{a}_i}$ is interpreted as a random sample drawn from the hashtag language model of h'. Note, this model does not consider preferences of the social network user leading to different annotation behaviour among users, this is in the scope of future work. Given a new document d' the model recommends k hashtags by

$$\mathit{rec}_k(d', \mathcal{M}_{\mathit{ALM}}) = \underset{\{h_1..h_k\} \subset \mathcal{H}}{\arg\max} \sum_{i=1}^{k} P(h_i|d')$$

This goes along the lines with the maximum a posteriori decision rule. We assume independence of the features (i.e., terms) and can omit the feature evidence, because we are interested in a ranking:

$$P(h|d') \propto P(h) \cdot \prod_{t \in d'} P(t|h)$$

The hashtag prior $P(h)$ is proportional to the frequency of alignments with hashtag h over all alignments $\mathcal{A}$ in the model. The term likelihood $P(t|h)$ can be estimated as the relative frequency of t in the language model document of h. To avoid numerical problems (i.e., for rare hashtags with small language models) we further apply Dirichlet prior smoothing ($\mu = 0.01$). Experiments showed superior quality in comparison to other smoothing strategies (i.e., Laplace and Jelinek-Mercer smoothing). An unary update performed on the model, i.e., *add* (or *rem*) of one alignment a, is restricted to exactly one hashtag (h_a). Hence, the class prior

[3] http://dev.twitter.com/streaming

[4] http://httparchive.org/interesting.php?l=Jan%201%202015

[5] http://lucene.apache.org/core/

[6] see http://lucene.apache.org/solr/ or http://www.elasticsearch.org

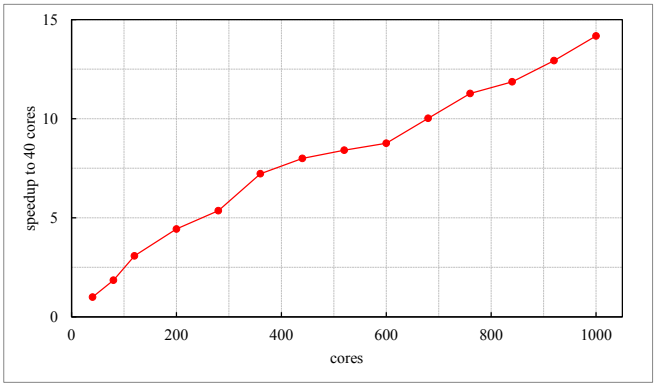


Figure 1: Speed-up of write operations in comparison to a 40-core configuration

is updated by incrementing (respectively decrementing) the alignment frequency of h_a by 1. Furthermore, the language model is updated by incrementing (or decrementing) the frequencies of all terms $t \in d_a$.

The *scalability* of ALM can be achieved in a straightforward manner. Recalling that updates of the models can be incorporated over different hashtags independently, the model can be distributed up to the point, where each node is responsible for one hashtag language model. Furthermore, the recommendation process can be parallelized in the same manner, because the calculation of the likelihood ($\prod P(t|h)$) is the decisive factor. We implemented a distributed version of ALM and achieved an almost linear speedup up to a computing cluster with 1,000 CPU cores[7]. Figure 1 depicts the relative speedup of the distributed version of ALM executed on different cluster configurations in comparison to a basic one node setup with 40 cores. The basic setup is already able to process approximately 130 alignments per second. Using 25-times more resources ($1,000$ cores), the throughput increased to approximately 1.866 alignments per second, which is a speedup of 14.2. While performing concurrent writes, ALM was capable of providing 150 recommendations per second. Both values show, that the distributed implementation of ALM is capable of dealing with the real-world data volumes produced by today's social network platforms.

[7]we used the resources provided by the HPI Future SOC Lab: http://hpi.de/en/research/future-soc-lab

4. EVALUATION

To evaluate the presented recommendation models, we prepared a dataset based on tweets retrieved from the Twitter Streaming API in a three month period (between August, 5^{th} and November, 4^{th} 2013). The Twitter API does not provide access to all tweets shared in the network, hence, we decided to retrieve only English tweets containing the keyword "http" to get as many as possible shared links. After removing tweets with no hashtags or inaccessible links (e.g., robots.txt restrictions or the HTTP code was not 200), the collection contained approximately 64M tweets, 6M hashtags and 12M URLs.

As shown in Figure 2, the hashtag usage follows a power-law like distribution. Because, we want to discover temporal effects in the hashtag usage, we removed hashtags used less than 20 times. This approximately corresponds to hashtags occurring less than once every 5^{th} day. Furthermore, we removed alignments with documents smaller than 900 characters (excluding boilerplate[8]). This was done because in a manual inspection we found that the majority of smaller texts contained many error messages (indicating that server or client showed unexpected behaviour) or only a minority of the actual page content. The properties of the resulting dataset are shown in Table 1.

Table 1: Dataset statistics

property	data
Number of tweets	12,303,814
Number of hashtags	63,506
Number of Web pages	4,595,843
Size of Web page contents	365.63 GiB
Number of alignments	19 306 419

To evaluate the recommendation quality, we follow the intuition, that all hashtags aligned by a user to a specific document should be considered as correct. Accordingly, all other hashtags (i.e., not shared in the context of the document) are considered to be incorrect recommendations. We define the set of ground-truth hashtags H_g for a document d' as the set hashtags in $\mathcal{H}$ aligned with the same document (i.e., same URL after resolving redirects due to URL shortener etc.):

$$H_g = \{h_{a_i} \in \mathcal{H} : d_{a_i} = d'\}$$

[8]http://code.google.com/p/boilerpipe/

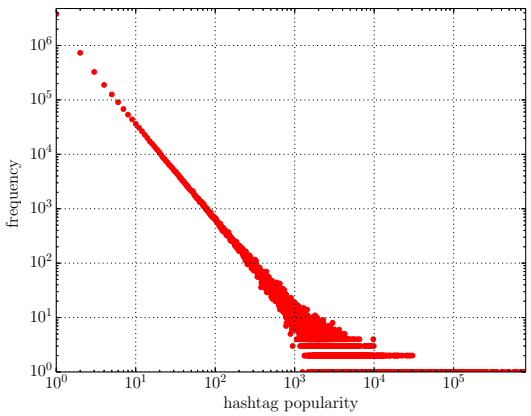


Figure 2: Hashtag popularity distribution

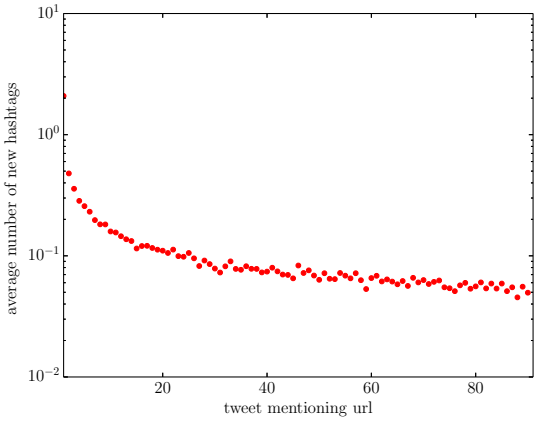
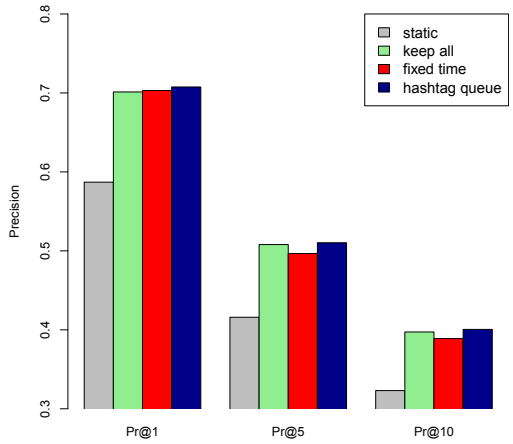
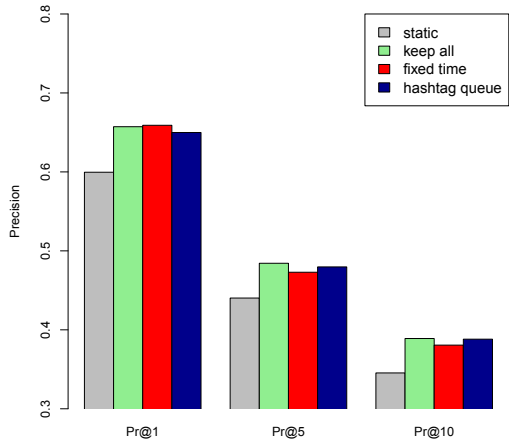


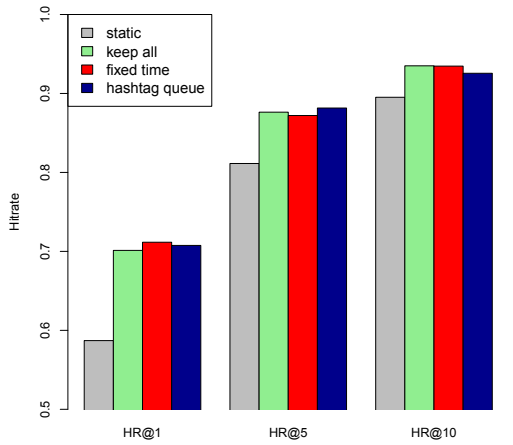
Figure 3: Average number of new hashtags for a document



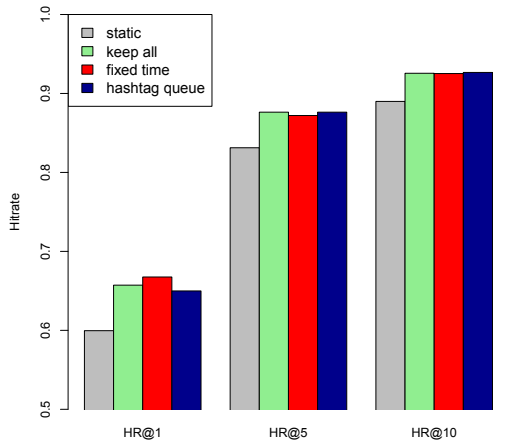
(a) Precision-at-k for HSD



(b) Precision-at-k for ALM



(c) Hitrate-at-k for HSD



(d) Hitrate-at-k for ALM

Figure 4: Qualitative evaluation results of the two models ALM and HSD w.r.t. precision- and hitrate-at-k

This conversely means: the more often a document is shared, the larger H_g. Figure 3 shows the ratio of new hashtags added to H_g per tweet (chronologically). The first tweet aligns a document with 2.09 hashtags on average, whereas after the 40^{th} tweet, only 5% of the subsequent tweets provide a new hashtag. Note, we consider only tweets with one or more hashtags here. This shows, that H_g converges early.

Given the ground truth H_g and the set of recommended hashtags $H_{\mathcal{M}} = rec_k(d', \mathcal{M})$, the recommendation quality is evaluated using precision-at-k ($Pr@k_{d'}$) to measure the precision within the top-k recommendation and hitrate-at-k ($HR@k_{d'}$) to indicate whether one of the top-k recommendations was correct:

$$Pr@k_{d'} = \frac{H_g \cap H_{\mathcal{M}}}{H_g} \quad HR@k_{d'} = \begin{cases} 1 & : \left|H_g \cap H_{\mathcal{M}}\right| > 0 \\ 0 & : else \end{cases}$$

Note, for $k = 1$ both values are equivalent ($Pr@1_{d'} = HR@1_{d'}$). Increasing values for k will lead to an decrease of precision but an increase of the hitrate.

To get a representative set of ground truth documents $\mathcal{D}$, we sampled 954 documents from the dataset. We limited the first sharing date of the documents between September 5^{th} and November 4^{th} to enable the same initial training phase for all compared models (1 month). Each document had to be aligned to 10 or more hashtags, because the top $k = 10$ documents should be evaluated. Finally, only documents from news pages[9] were considered for the evaluation. We argue that not all frequently tweeted Web pages are suited for evaluation purposes, because many advertisements are frequently tweeted with specific hashtags. For instance, App Store pages of mobile games, like "Smurfs' Village"[10], are often annotated with proper nouns specific to the game (e.g., #SmurfsVillage). This makes it easy to predict relevant hashtags for these Web pages because the specific nouns are only used in the context of the game's hashtags.

To estimate the recommendation quality over several documents

[9]We used a list of news sources applied by Google: News http://labnol.org/tech/google-news-sources

[10]http://itunes.apple.com/app/id399648212

$\mathcal{D}$, we build the average over all documents of all precision@k (or hitrate@k).

Besides the introduced update strategies fixed time and hashtag queue we evaluate two base lines. First, we show the performance of recommendation models with no update strategies (following referred to as "static"). These models are trained based on the initial training alignments (i.e., the alignments shared between August 5^{th} and September 5^{th}) and are following not updated. They build a baseline recommender with no further knowledge about current developments in Twitter. Second, we show the performance of a "keep all" model, that does not remove any alignments. As previously discussed, such a model would not work in practice, because it will grow infinitely. However, it shows the performance of a recommender with the maximal possible knowledge. All four strategies are applied for the two recommendation models, i.e., the document similarity based approach HSD and the probabilistic approach of ALM.

Figures 4a and 4b present the precision-at-k values for different update strategies for of HSD and ALM, where Figures 4c and 4d oppose the hitrate-at-k of both models based on the update strategies. Overall, all configurations yield notable good results. Recalling, that the solved problem is to recommend 1 (respectively 5 or 10) out of 63,506 hashtags for a given document. For instance, the hitrate-at-10 results show that for over 90% of the documents one ore more appropriate hashtags were correctly recommended. The precision results underline that. Considering 10 recommended hashtags for a single document, on average three or more correct hashtags were recommended by all configurations.

Furthermore, the application of an update strategy strongly influences on the recommendation results. The static configuration produces between 4 and 12 percent point lower precision and hitrate values for both recommendation models. This underlines the effectiveness of the update strategies and underlines, that the contexts shared via Twitter underlie a constant change. The differences in recommendation quality for the ALM are smaller. Furthermore, HSD outperforms ALM by 1 to 5 percent points in terms of precision-at-k. This shows that HSD adopts better to current changes of the context drifts of hashtags. This is because already one alignment of a similar document might influence the recommendations of HSD significantly. In contrast to that, ALM requires a considerable change of a hashtags language model.

The differences between the introduced update strategies (fixed time and hashtag queue) and the "keep all" baseline are small. The variations of precision and hitrate vary in a range of ± 1 percent points and can not justify a significant difference of the strategies. This shows that, despite based on "more" data, the keep all strategy does not outperform the other strategies. It remains to show, how a dataset over a longer time windows would influence the quality of the recommendations (e.g., one year). We expect that an evaluation over a longer time span might show a superior recommendation quality of the the introduced update strategies (fixed time and hashtag queue).

5. CONCLUSION

In this work, we investigate the problem of tag recommendation for documents shared over the social network Twitter. We argue, that the tagging behaviour of the Twitter community underlies strong temporal dynamics such as trending topics. We introduce two basic update strategies for hashtag recommendation models that help to adopt to the current developments of the sharing behaviour of the community. Furthermore, we discuss two basic recommendation strategies compatible to the introduced update strategies. One strategy is based on the observation of tagging behaviour is akin among similar documents. The second strategy is based on the assumption, that each hashtag represents a word distribution from which the aligned documents are generated.

In future work, we want to investigate more sophisticated recommendation models that consider inter-hashtag-dependencies and varying tagging behaviour among users. Furthermore, more evolved update strategies have to be elaborated. For instance, irrelevant alignments in the model might be detected, iff new incoming alignments can not be explained with them.

6. ACKNOWLEDGEMENTS

This research was funded by the German Research Society (DFG grant no. FOR 1306 - http://stratosphere.eu/).

7. REFERENCES

[1] P. A. Chirita, S. Costache, W. Nejdl, and S. Handschuh. P-tag: Large scale automatic generation of personalized annotation tags for the Web. In *Proceedings of WWW*, pages 845–854, 2007.

[2] S. A. Golder and B. A. Huberman. Usage patterns of collaborative tagging systems. *Journal of information science*, 32(2):198–208, 2006.

[3] M. Gupta, R. Li, Z. Yin, and J. Han. Survey on social tagging techniques. *SIGKDD Explorations*, 12(1):58–72, Nov. 2010.

[4] B. Han and T. Baldwin. Lexical normalisation of short text messages: Makn sens a #Twitter. In *Proceedings of ACL*, pages 368–378, 2011.

[5] M. Hu, S. Liu, F. Wei, Y. Wu, J. Stasko, and K.-L. Ma. Breaking news on Twitter. In *Proceedings of SIGCHI*, pages 2751–2754, 2012.

[6] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme. Tag recommendations in folksonomies. In *Proceedings of PKDD*, pages 506–514, 2007.

[7] R. Krestel, P. Fankhauser, and W. Nejdl. Latent Dirichlet allocation for tag recommendation. In *Proceedings of RecSys*, pages 61–68, 2009.

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

[9] S. M. Kywe, T.-A. Hoang, E.-P. Lim, and F. Zhu. On recommending hashtags in Twitter networks. In *Proceedings of SocInfo*, pages 337–350, 2012.

[10] J. Lehmann, B. Gonçalves, J. J. Ramasco, and C. Cattuto. Dynamical classes of collective attention in Twitter. In *Proceedings of WWW*, pages 251–260, 2012.

[11] A. Mazzia and J. Juett. Suggesting hashtags on Twitter, 2009.

[12] S. Sedhai and A. Sun. Hashtag recommendation for hyperlinked tweets. In *Proceedings of SIGIR*, pages 831–834, 2014.

[13] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web (TWEB)*, 5(1):4:1–4:31, 2011.

[14] E. Zangerle, W. Gassler, and G. Specht. On the impact of text similarity functions on hashtag recommendations in microblogging environments. *Social Network Analysis and Mining*, 3(4):889–898, 2013.

[15] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*, 22(2):179–214, 2004.