MLJ: Language-Independent Real-Time Search of Tweets Reported by Media Outlets and Journalists

Masumi Shirakawa, Takahiro Hara, Shojiro Nishio
Graduate School of Information Science and Technology, Osaka University, Japan
{shirakawa.masumi, hara, nishio}@ist.osaka-u.ac.jp

ABSTRACT

In this demonstration, we introduce MLJ (MultiLingual Journalism, http://mljournalism.com), a first Web-based system that enables users to search any topic of latest tweets posted by media outlets and journalists beyond languages. Handling multilingual tweets in real time involves many technical challenges: language barrier, sparsity of words, and real-time data stream. To overcome the language barrier and the sparsity of words, MLJ harnesses CL-ESA, a Wikipedia-based language-independent method to generate a vector of Wikipedia pages (entities) from an input text. To continuously deal with tweet stream, we propose one-pass DP-means, an online clustering method based on DP-means. Given a new tweet as an input, MLJ generates a vector using CL-ESA and classifies it into one of clusters using one-pass DP-means. By interpreting a search query as a vector, users can instantly search clusters containing latest related tweets from the query without being aware of language differences. MLJ as of March 2014 supports nine languages including English, Japanese, Korean, Spanish, Portuguese, German, French, Italian, and Arabic covering 24 countries.

1. INTRODUCTION

The mainstream of journalism has moved to social media including Twitter and Facebook where media outlets and journalists can report and disseminate their own stories in real time. According to the Cision's report on social journalism[1], approximately 90 percent of journalists in UK, France, and Canada use microblogs for their work. Surprisingly, Germany's 59 percent is the lowest among surveyed nine countries. This indicates that social journalism has become common in not a few countries. There are already some curation services that focus on tweets posted by media outlets and journalists, e.g. Muck Rack[2] and Lissted[3].

While social journalism has been spreading across the globe, the language barrier still remains as a big problem. In Muck Rack and Lissted, it is difficult to find tweets posted from Non-English speaking countries. However, knowing stories reported in other countries is important to understand the actual mood of the countries. Sáez-Trumper et al. [5] revealed that online media outlets in a certain country tend to select the same stories. This implies that stories reported in a country may be unique and worth spreading in other countries. Practically, Arabic tweets from within Egypt were translated and retweeted to notify non-Arabic speakers of the actual mood during Arab Spring [1].

A few work has attempted to build a system that is capable to search news stories across languages. A representative work is Europe Media Monitor (EMM)[4] [7] developed by European Commission. NewsExplorer, a main function of EMM, provides multilingual Web news search among 19 languages. It utilized cross-language links of Wikipedia to bridge the gap among languages. Columbia Newsblaster[5] [2] tried to handle multilingual Web news using machine translation techniques, though the system does not provide multilingual function as of March 2014. To the best of our knowledge, there is no system that enables users to search tweets posted by media outlets and journalists across languages.

In this demonstration, we present MLJ (MultiLingual Journalism), a system to find latest related tweets written in any language reported by media outlets and journalists. Using MLJ, users can search tweets of coverage using a text query without being aware of the difference of languages. The remains of this paper are organized as follows. Key technologies of MLJ are described in Section 2, followed by the system architecture of MLJ in Section 3. In Section 4, we illustrate what will be demonstrated in the conference. Finally, we conclude our work in Section 5.

2. KEY TECHNOLOGIES

2.1 Relatedness Measurement Between Texts Across Languages

The language barrier and the sparsity of words are the most challenging problems for finding related tweets across languages. In order to bridge the language barrier, the unification of language spaces should be accomplished. Also, the enhancement of information amount is needed to overcome the sparsity of words. To solve both problems, we

[1] http://us.cision.com/thought-leadership/2013-social-journalism/
[2] http://muckrack.com/
[3] http://lissted.com/
[4] http://emm.newsbrief.eu/overview.html
[5] http://newsblaster.cs.columbia.edu

Proceedings of the VLDB Endowment, Vol. 7, No. 13

Algorithm 1 ESA

Input: text T, language L
Output: feature vector $\boldsymbol{V}_L \in \mathbb{R}^{|E_L|}$

$\boldsymbol{V}_L = \mathbf{0}$
for each term $t_i \in T$ **do**
  for each entity $e_j \in E_L$ **do**
    $V_L[e_j] = V_L[e_j] +$ TermScore(t_i, e_j, L)
  end for
end for

Algorithm 2 CL-ESA

Input: text T, language L, base language B
Output: feature vector $\boldsymbol{V}_B \in \mathbb{R}^{|E_B|}$

$\boldsymbol{V}_B = \mathbf{0}$
$\boldsymbol{V}_L =$ESA(T, L)
for each entity $e_j \in E_L$ **do**
  if $e_j \in E_B$ **then**
    $V_B[e_j] = V_L[e_j]$
  end if
end for

employ Cross-Lingual Explicit Semantic Analysis (CL-ESA) [6]. CL-ESA is based on Wikipedia-based Explicit Semantic Analysis (ESA) [3], which generates a vector of Wikipedia pages (entities) from an input text. ESA specifically builds an inverted index that maps each word into a vector of Wikipedia pages in which it appears. ESA can expand the semantic information of a tweet to address the problem of the sparsity of words. CL-ESA extends ESA using inter-language links of Wikipedia to generate a vector of any language version of Wikipedia pages. We use CL-ESA to generate a vector of English Wikipedia pages from a tweet written in any language.

Algorithm 1 represents a pseudo-code of ESA. Given text T and its language L as the input, it computes the score of term $t_i \in T$ for each Wikipedia page of language L describing entity e_j (e_j indicates a unique entity in the world and is independent of the language) using TFIDF or BM25 (TermScore), and sums the term scores for each e_j. The output is a feature vector $\boldsymbol{V}_L$ whose dimensional space is $\mathbb{R}^{|E_L|}$. E_L is the entity space of Wikipedia of language L.

Algorithm 2 describes a pseudo-code of CL-ESA. It generates a feature vector of Wikipedia pages of language B (we set B as English in this work) from input text T of language L. It utilizes the output of ESA, i.e., it converts a feature vector of Wikipedia pages of language L into that of Wikipedia pages of language B. If and only if a Wikipedia page of language L describing entity e_j has a cross-language link to a Wikipedia page of language B, value $V_L[e_j]$ is copied to $V_B[e_j]$. Consequently, text T of any language L is interpreted as a feature vector $\boldsymbol{V}_B$ in unified dimensional space $\mathbb{R}^{|E_B|}$. Relatedness between texts written in different languages are computed using $\boldsymbol{V}_B$.

2.2 Online Clustering for Stream Data

MLJ sorts related tweets using a clustering method in order to make tweet stream quickly searchable. Here, we introduce an online clustering method based on DP-means algorithm [4]. DP-means does not need to determine the number of clusters beforehand. This characteristic is suited to Twitter data where the number of topics cannot be determined. However, original DP-means is a batch clustering algorithm which cannot handle stream data. We therefore propose one-pass DP-means, an online clustering algorithm based on DP-means. Note that our algorithm belongs to spherical clustering which employs cosine similarity instead of Euclid distance.

Algorithm 3 One-pass DP-means

Input: new data d, threshold λ, clusters $C_1, \cdots, C_K$
Output: clusters $C_1, \cdots, C_K$ (K may be incremented)

$s_{max} = \max(\text{Cosine}(d, C_k), k = 1, \cdots, K)$
$k_{max} = \arg\max_k(\text{Cosine}(d, C_k), k = 1, \cdots, K)$
if $s_{max} > \lambda$ **then**
  $C_{k_{max}} = C_{k_{max}} \cup \{d\}$
else
  $C_{K+1} = \{d\}$
  $K = K + 1$
end if

Algorithm 4 Related cluster search in VSM

Input: text query q, threshold λ_q
Output: a set of related clusters A

$A = \phi$
for each cluster C_k **do**
  $s = \text{Cosine}(q, C_k)$
  if $s > \lambda_q$ **then**
    $A = A \cup \{C_k\}$
  end if
end for

Algorithm 3 describes a pseudo-code of one-pass DP-means. The inputs of the algorithm are new data d, threshold λ, and existing clusters $C_1, \cdots, C_K$. It first calculates the cosine similarity between d and each cluster C_k, and finds the highest similarity s_{max} with its id k_{max}. If s_{max} exceeds threshold λ, it adds d to cluster $C_{k_{max}}$. Otherwise it generates a new cluster C_{K+1} and adds d to C_{K+1}. Namely, the new data d is added to either the most closest cluster or a new empty cluster. Existing data do not need to be moved into other clusters in this algorithm. In our system, data d is a tweet and a cluster C_k is a set of tweets. Cosine similarity is computed using $\boldsymbol{V}_B$ generated from each tweet.

2.3 Search in Vector Space Model

Since all tweets and clusters are mapped into the vector space of $\mathbb{R}^{|E_B|}$, we introduce a search technique for vector space model (VSM). Specifically, a search query is also interpreted as a feature vector $\boldsymbol{V}_B$. This enables users to search related tweets by a query written in any language.

Algorithm 4 describes a pseudo-code of related cluster search in VSM. The inputs are text query q and threshold λ_q. Related cluster search is performed by similar approach to one-pass DP-means, i.e., it computes the cosine similarity between q and each cluster C_k, and enumerates all clusters whose similarities are above λ_q. A, a set of related clusters containing related tweets, is returned as the output. λ_q adjusts the trade-off of the accuracy and coverage of related tweets.

3. SYSTEM ARCHITECTURE

We incorporated key technologies explained in Section 2 to develop MLJ. It mainly consists of two components, i.e.,

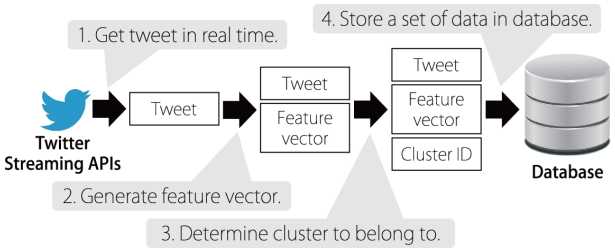

Figure 1: Tweet Collector.

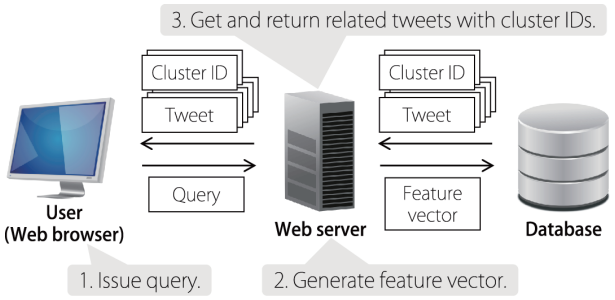

Figure 2: Query Processor.

Tweet Collector and Query Processor. Figures 1 and 2 respectively illustrate the processes of Tweet Collector and Query Processor.

3.1 Tweet Collector

Using User Streams of Twitter Streaming APIs[6], Tweet Collector continuously obtains tweets posted by whitelisted Twitter accounts of media outlets and journalists. To find media outlets and journalists on Twitter, we used Twitter's *who to follow* function[7]. Specifically, we collected 613 accounts speaking nine languages (English, Japanese, Korean, Spanish, Portuguese, German, French, Italian, and Arabic) from 24 countries. The account list can be found at our Twitter account *@mljournalism*[8]. Because our approach is language-independent, we consider supporting more languages in the future.

Continuously obtained tweets are immediately processed one by one and stored in a database. To map each tweet of any language into the same dimensional space, the system generates a vector of English Wikipedia pages using CL-ESA (Section 2.1). Since CL-ESA requires the language of a tweet as the input, it uses CLD2 (Compact Language Detection 2)[9] to detect the language[10]. If the language of a tweet is detected as English, the system applies ESA to the tweet. Here, English Wikipedia pages that have no cross-language link are ignored for better comparison with feature vectors generated from other languages. The maximum number of non-zero elements in a feature vector is set as 200 to keep it sparse, saving the computational cost and storage space. Each vector is then normalized. It is noteworthy that the text part of a tweet is extracted before applying CL-ESA. URLs, mentions (i.e., "@" plus user ID), retweet symbols (i.e., "RT," "QT," and "MT" before mentions), and fixed phrases are removed from a tweet. In order to determine fixed phrases for each Twitter account, we collected latest 100 tweets and extracted phrases appearing more than half of the tweets.

After generating a feature vector of a tweet, online clustering is performed. One-pass DP-means (Section 2.2) is specifically applied to the feature vector to determine either adding the tweet into the most closest cluster or generating a new cluster. The system does not update the feature vector of an existing cluster when adding the tweet into it. This is because updating the inverted index for each tweet requires much computational time. When generating a new cluster, i.e., when the tweet is not close to any of existing clusters, the feature vector of the cluster is copied from that of the tweet. We set the threshold λ of cosine similarity as 0.25 to avoid a single cluster containing different topics of tweets. Since the maximum number of non-zero elements is set as 200 among millions of dimensions (i.e., the number of English Wikipedia pages having at least one cross-language link), the situation that the cosine similarity between a tweet and a cluster exceeds 0.25 indicates they are very similar. The tweet, the feature vector, and the cluster ID are finally stored in the database. Once a set of data is stored in the database, there is no need to update it.

3.2 Query Processor

While Tweet Collector stores tweets posted by media outlets and journalists in real time, Query Processor handles queries issued by MLJ users via Web browsers. In order to achieve language-independent search of tweets, queries are also interpreted as feature vectors of English Wikipedia pages as well as tweets and clusters. In the same dimensional space, all related clusters that contain related tweets are retrieved using search techniques in VSM (Section 2.3). The threshold λ_q for queries can be set by users to adjust the trade-off of the accuracy and coverage. As λ_q is decreased, clusters obtained from a query cover more related tweets while sacrificing the accuracy. The system returns a limited number of latest related tweets with cluster IDs as of when the query is issued. Older tweets can be retrieved by using cluster IDs that are obtained at the first time.

4. DEMONSTRATION

In our demonstration, we show the potential of MLJ to find latest stories of various languages and countries on Twitter. We use the system described in Section 3 that is continuously storing tweets in real time. Figure 3 represents an example of use of MLJ. A search box is located in the top left of Figure 3. A Japanese query indicating "Malaysia airplane" was issued at 2014-03-24 03:35 (GMT). In the right, tweets related to "Malaysia airplane" posted by each country's media outlets and journalists were displayed as a timeline. As of March 2014, Twitter officially supports translation function using Bing Translator[11]. Users who have their own Twitter account can read tweets in their own language by accessing official Twitter page (clicking "TWEET" in Figure 3). The timeline by default shows latest 30 related tweets on the Web browser. Users are able to obtain older tweets by scrolling down to the bottom.

[6]http://dev.twitter.com/docs/streaming-apis
[7]http://twitter.com/who_to_follow/interests
[8]http://twitter.com/mljournalism
[9]https://code.google.com/p/cld2/
[10]Twitter officially provides language attributes of users but they do not always agree with actual languages of tweets.
[11]http://www.bing.com/translator

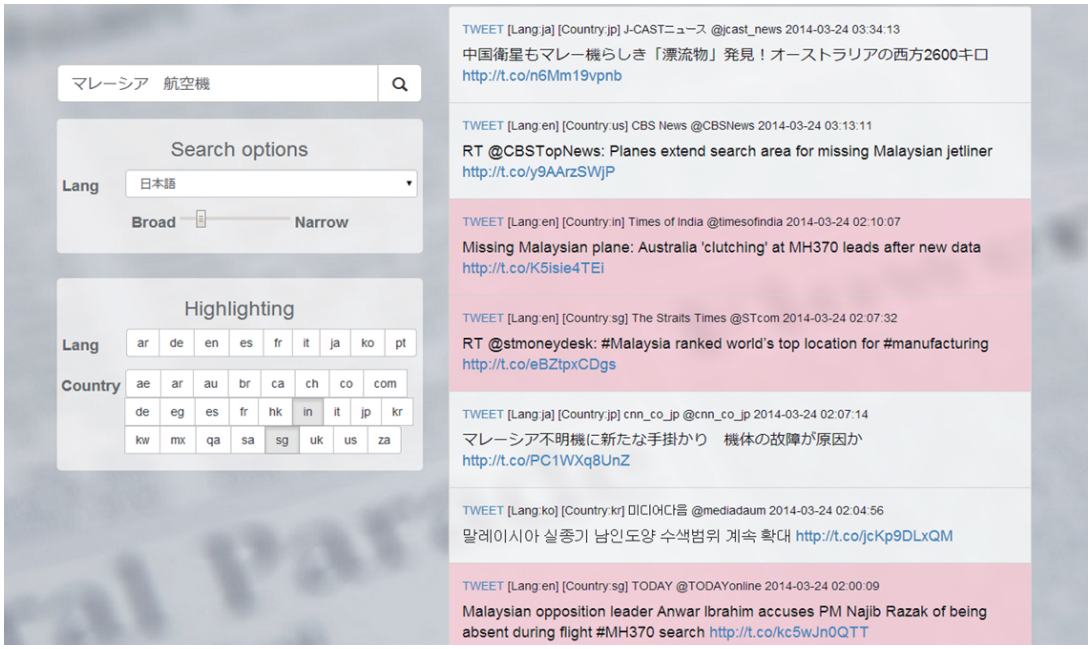

Figure 3: An example of use of MLJ (http://mljournalism.com). Japanese query indicating "Malaysia airplane" was issued at 2014-03-24 03:35 (GMT). Latest tweets related to "Malaysia airplane" posted by media outlets and journalists in various countries were displayed as a timeline.

Search options are shown under the search box in the left of Figure 3. As the search option, users can select the language of search queries (the default language is set using browser settings). Users speaking any of supported nine languages can use MLJ to search any topic of tweets written in any of the languages. Also, users can adjust the threshold λ_q of the similarity as the other search option. λ_q can be set between 0.01 (Broad) and 0.07 (Narrow) using the slider.

Highlighting options are placed under the search options in the left of Figure 3. They enable users to highlight tweets by languages or countries. In Figure 3, "in" (means India) and "sg" (means Singapore) buttons were turned on. Tweets posted from within India and Singapore were highlighted as red in the timeline. Using the highlighting options, users can target tweets of specific languages or countries.

The demonstration will be interactively conducted at the conference. That is, one can freely use MLJ to search stories of interest at that time. Since Figure 3 shows a snapshot on 2014-03-24, it cannot be displayed in the demonstration. The system returns different search results for the same query depending on the time the query is issued. One will obtain the latest tweets related to the query. To show the ability of MLJ to catch up current stories around the world, we will also demonstrate several examples using sample queries reflecting the current affairs. Figure 3 is an example of catching up the accident of Malaysia Airlines Flight 370 which occurred on 2014-03-08.

5. CONCLUSION

In this paper, we introduced MLJ (MultiLingual Journalism), a Web-based system that enables users to search tweets posted by media outlets and journalists beyond languages. The future work includes supporting more languages such as Chinese and Russian, detecting popular news stories in each country or across the border, and evaluating the precision and recall of search results and the processing time.

6. ACKNOWLEDGMENTS

This work was supported in part by CPS-IIP Project (Integrated Platforms for Cyber-Physical Systems to Accelerate Implementation of Efficient Social Systems (FY2012 - FY2016)) in the research promotion program for national level challenges "research and development for the realization of next-generation IT platforms" by the Ministry of Education, Culture, Sports, Science and Technology (MEXT).

7. REFERENCES

[1] M. Aouragh and A. Alexander. The Egyptian Experience: Sense and Nonsense of the Internet Revolution. *International Journal of Communication*, 5:1344–1358, 2011.

[2] D. K. Evans, J. L. Klavans, and K. R. McKeown. Columbia Newsblaster: Multilingual News Summarization on the Web. In *HLT-NAACL: Demonstration Papers*, pages 1–4, May 2004.

[3] E. Gabrilovich and S. Markovitch. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In *IJCAI*, pages 1606–1611, Jan. 2007.

[4] B. Kulis and M. I. Jordan. Revisiting k-means: New Algorithms via Bayesian Nonparametrics. In *ICML*, pages 513–520, July 2012.

[5] D. Sáez-Trumper, C. Castillo, and M. Lalmas. Social Media News Communities: Gatekeeping, Coverage, and Statement Bias. In *CIKM*, pages 1679–1684, Oct./Nov. 2013.

[6] P. Sorg and P. Cimiano. Cross-lingual Information Retrieval with Explicit Semantic Analysis. In *Working Notes for the CLEF 2008 Workshop*, Sept. 2008.

[7] R. Steinberger, B. Pouliquen, and E. van der Goot. An Introduction to the Europe Media Monitor Family of Applications. In *SIGIR Workshop on Information Access in a Multilingual World*, pages 1–8, July 2009.