Time-Independent and Language-Independent Extraction of Multiword Expressions From Twitter

Nikhil Londhe
SUNY Buffalo
nikhillo@buffalo.edu

Rohini K Srihari
SUNY Buffalo
rohini@buffalo.edu

Vishrawas Gopalakrishnan
SUNY Buffalo
vishrawa@buffalo.edu

Abstract

Multiword Expressions (MWEs) are crucial lexico-semantic units in any language. However, most work on MWEs has been focused on standard *monolingual* corpora. In this work, we examine MWE usage on Twitter - an inherently multilingual medium with an extremely short average text length that is often replete with grammatical errors. In this work we present a new graph based, language agnostic method for automatically extracting MWEs from tweets. We show how our method outperforms standard Association Measures. We also present a novel unsupervised evaluation technique to ascertain the accuracy of MWE extraction.

1 Introduction

Apart from being just a social media platform, Twitter has emerged as an authoritative source of breaking news and subsequent discussions (Kwak et al., 2010; Hu et al., 2012). Most “global” news stories, from terrorist attacks, political news, sports events to celebrity updates, not only *trend* on Twitter within minutes of the actual event but often in multiple languages. One challenge thus, in understanding the full story is being able to process all languages involved. One way to do this could be by partitioning data into the constituent languages (Bergsma et al., 2012) as there exist several sophisticated tools for Twitter (Pak and Paroubek, 2010; Ritter et al., 2012; Owoputi et al., 2013; Kong et al., 2014) designed specifically for various languages (Avontuur et al., 2012; Abdul-Mageed et al., 2012; Rehbein, 2013). However, such an approach might not be able to process all languages. Further, it faces an added disadvantage of ignoring valuable semantic, temporal and cross-lingual relationships between the tweets. In fact these relationships could instead be utilized to not only better understand the underlying story but also generate resources for resource poor languages in question.

Thus, as a cursory step in understanding such hashtags, our work focuses on extracting multiword expressions (MWEs) from Twitter data streams. MWEs are great starting points from two perspectives: (a) they are statistically “idiosyncratic” (Sag et al., 2002) and thus, require no prior knowledge of the text or the corresponding language for extraction and (b) form a considerable portion of the vocabulary for a given language (Fellbaum, 1998). Furthermore, their importance for a variety of NLP tasks like POS tagging (Shigeto et al., 2013), deep parsing (Nivre and Nilsson, 2004), sentiment analysis (Moreno-Ortiz et al., 2013), translation (Ren et al., 2009; Carpuat and Diab, 2010) etc. cannot be overstated. Also, as we explore in Section 4, MWE usage on Twitter shows some unique characteristics stemming from the nature of the medium like acronym usage, temporal sensitivity, etc. and thus, motivating a stronger need to develop MWE extraction techniques specific to such data streams.

However, most work (Van de Cruys and Moirón, 2007; Ramisch et al., 2010; Sinha, 2011) on automatic MWE extraction has either relied on (a) the knowledge of POS patterns that constitute MWEs and the availability of POS annotated corpora, or (b) enumeration of all possible n-grams and ranking them using Association Measures (AMs) (Pedersen et al., 2011). A third branch of work also exists that instead uses parallel corpora (Da Silva et al., 1999) and exploits distributional dissimilarity between words

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 2269–2278, Osaka, Japan, December 11-17 2016.

S. no	Tag type	POS tags	Examples
1	ADJP	JJ JJ	*petits blancs*, ginger redhead
2		NN JJ	day gay, reunion special
5	NP	JJ NN	*delicioso cctel*, *sozialen netzwerken*
6		DT NN	*la eurocopa*
7		NN NN	clapback season, skai jackson, **fra rou**, *asie pacifique*
8		NN RB	*la arranca*
9	ADVP	PP RB	*mal den*
10	VP	NN VB	*je suis*, *kuch lana*, *nahi aayenge*
11		RB VB	verbally attacked, *heit aber*
12		VB JJ	breaking federal
13		VB NN	cry blood, banish demons, minimize disruption, *evitar el*
14		VB RB	starts tonight, acted honorably
15		VB VB	gotta catch, **lets rt**

Table 1: Examples of extracted MWEs and their syntactic classification

to extract phrases. However, we do not consider this approach further given the target domain and only mention it here for completeness.

However, as outlined above, the very nature of our problem invalidates the first line of approach. It is impractical to build corresponding systems (namely POS taggers, POS patterns and candidate extraction) for every applicable language. As far as the second approach is concerned, it is usually effective over time invariant datasets where one time enumeration of all n-grams would suffice. However, our setting would require frequent regeneration of N-grams as the corpus increases over time. Hence, we would like to find methods that do not require enumerating all N-grams and can yet find statistically significant phrases. An added challenge, as we discuss in Section 2, when working with multilingual data is that of evaluation. Thus, we must also find ways to evaluate the extracted MWEs that involves minimal manual intervention.

Thus, the primary objectives of this work can be enumerated as:

- Propose a new graph based method for MWE extraction that can circumvent the challenges of Twitter language usage, temporal nature of Hashtags and possible enumeration of all N-grams.
- Propose an automatic evaluation technique for the extracted MWEs
- Additionally, analyze the variance in extracted MWEs across different variables

The rest of the paper is organized as follows. Starting with Section 2, we first discuss the problem setting in a little more detail and then present our method in Section 3. We show why a word graph based method can overcome the enumerated problems - multilingualism, lack of grammar and relatively free word ordering to name a few. Then in Section 4, we describe our novel evaluation technique and also compare the performance of our method against different AMs. Finally, we conclude by discussing the scope of future work and conclusions from our results in Section 5.

2 Related Work

In this section, we consider the problem of extracting MWEs from a text corpus and evaluating the accuracy and nature of the extracted MWEs. As discussed in Section 1, nuanced extraction techniques rely on POS annotated corpora at the very least. Firstly, the lack of POS taggers for all applicable languages would reduce the size of the workable dataset. For example, some resource poor languages like Malay, Indonesian etc. have very little work in the said regard (Adriani and Van Rijsbergen, 2000; Rais et al., 2011). Secondly, as shown by (Derczynski et al., 2013), POS taggers trained on longer documents perform poorly on tweets. Further the extraction patterns vary widely (Kunchukuttan and Damani, 2008; Green et al., 2011; Tsvetkov and Wintner, 2014) based on the underlying language and thus, making it computationally intractable. Finally, as shown by (Solorio et al., 2014), it is much harder to detect the individual languages within code switched short text documents. Further, there is even lack of availability of standard annotated corpora beyond a handful of languages (Solorio and Liu, 2008; Vyas et al., 2014) for code switched text and thus, almost little to no research even exists in extracting MWEs from such text. Thus, at the very least we need to look at techniques that do not rely on POS tags.

As far as candidate evaluation is concerned, most techniques discussed thus far focus on evaluating the *efficacy* of the *POS extraction patterns*. Hence, a common technique (Pearce, 2002; Ramisch et al., 2012) involves measuring recall against standard corpora. Note that such methods assume that an exhaustive language specific list of MWEs is available. However, since our task is primarily concerned with MWE "discovery", such standard lexicons may not be used. A common alternative involves manual evaluation. However, our initial efforts at manual evaluation proved to be tedious primarily due to unfamiliarity with some of the languages. This prompted us to develop an automatic evaluation technique that we present in Section 4.2 that uses the Twitter Search API.

However, this raises a related yet contrary question on MWE classification. For the extracted MWEs to be useful for downstream processing, some nomenclature must be developed. Some of the earliest work in MWE extraction and classification was done by Sag et al. (2002). They initially introduced a structural classification for MWEs that relies on the differences in compositionality and fixedness between the different MWEs. Later work by Schneider et al. (2014) on MWE usage in social media uses two classification schemes. One, that deals with compositionality and classifies MWEs as either *strong* or *weak* based on their opaqueness and a second, detailed syntactic classification that relies on POS tags. In a multilingual scenario however, it is much easier to determine POS tags for a foreign phrase than to judge the compositionality or opaqueness of the MWE itself. Thus, in continuation with the list provided by Schneider et al. (2014) that deals specifically with social media, we adopted an abridged version[1] as depicted in Table 1. The table lists the tag type, the POS tags used and some extracted examples. Note that this scheme is used only for the purpose of classification and not utilized for MWE evaluation. For languages other than English, we determine membership by examining the translation of the given foreign language phrase. We largely use this nomenclature for analysis as presented in Section 4.

Having thus presented an overview of related work, we now turn our attention to our main algorithm.

3 System description & algorithms

3.1 Constructing Word Graphs

Thus, so far we have established that the nature and size of tweets are an hindrance for the standard tokenization process. However, using word graphs would circumvent both problems. On one hand, they would allow us to capture co-occurrence and statistical information within the graph structure but at the same time allow relaxed word ordering. Thus, given a set of tweets for a hashtag, which we will refer to as a *dataset*, we could construct a single graph $G = (V, E)$ from all tweets as follows. The vertices V represent the set of all unique tokens that occur within the dataset and two vertices share an edge if they co-occur within a tweet. The edge weight is set to the co-occurrence probability of the participating vertices and each vertex is annotated with the occurrence probability of the underlying token. The token set is obtained by simple whitespace tokenization followed by lowercasing and removing all mentions, URLs, emojis/emoticons and # prefixes.

For such a graph, we further contend that the tokens represented by a pair of vertices constitute a MWE if (a) the said tokens frequently co-occur but (b) rarely occur with other tokens. This could be ascertained by using the edge weights and examining the vertex neighborhoods of the said vertices. To that end, we looked at similar problems in other domains and found the method as presented by Londhe et al. (2014) for *Product title matching* to be promising. The authors essentially demonstrate how word graphs for product titles can be utilized to detect equivalences using a community detection algorithm viz. CDAM (Community Detection for Approximate Matching). We thus implemented equivalent algorithms, collectively called GRePE (**G**raph **Re**duction for **P**hrase **E**xtraction) in our problem setting which we now present.

3.2 Extracting MWE candidates

A block diagram of our system components is shown in Figure 1. Overall, the two main system components are the Indexer and the Graph Reducer. The *Indexer* ingests a given dataset to convert it into

[1] we only use bigrams and treat proper nouns as any other nouns

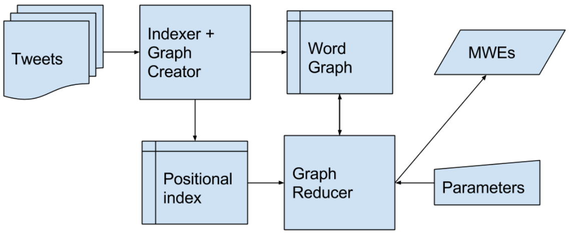


Figure 1: System block diagram

Algorithm 1 CROSS-VERTEX ENRICHMENT

```
Input: Pair of vertices $V_1$, $V_2$ and enrichment threshold $\eta$
Output: Enriched neighborhoods $\mathrm{N}^e(V_1), \mathrm{N}^e(V_2)$
for Vertex $v$ in $\{V_1, V_2\}$ do
    Find $v'$ in $\mathrm{N}(v)$ such that $p(v, v')$ is largest
    Define $\mathrm{N}_\eta(v)$ as all vertices $x$ with $p(x, v) \geq \eta \times p(v, v')$
    Initialize $\mathrm{N}^e(v) = \mathrm{N}_\eta(v)$
    Let $v_o = V \setminus v$ , $\mathrm{N}'_\eta(v) := \mathrm{N}(v) \setminus \mathrm{N}_\eta(v)$, $\mathrm{C}(v) = \mathrm{N}'_\eta(v) \cap v_o$
    Let $S_v = getWJC(\mathrm{N}(v), \mathrm{N}(v_o))$
end for
for Each element $c$ in $\mathrm{C}_1$ do
    Let $S_c = getWJC((\mathrm{N}(1) \setminus c), \mathrm{N}(2))$
    if $w_c = p(V_1, c) + |S_c - S_1| \geq \eta_1$ then
        Add $c$ to $\mathrm{N}^e_1$
    end if
end for
Repeat above for $C_2$
Return $\mathrm{N}^e(V_1), \mathrm{N}^e(V_2)$
```

a *Word Graph* and a corresponding *Positional Index*. The *Graph Reducer* then iterates over the graph, detects MWEs and merges constituent nodes. The following subsections present more details.

Before we describe the graph reduction algorithms, we introduce some notation as follows:

1. i^{th} vertex is denoted as V_i
2. The neighborhood of a vertex V, i.e. a set of vertices up to a depth of k, is denoted as $\mathrm{N}_k(V)$
3. Immediate neighborhood of a vertex V i.e. $\mathrm{N}_1(V)$ is denoted simply as $\mathrm{N}(V)$
4. $p(V)$ and $p(V_i, V_j)$ represent the prior and joint probabilities respectively

The process of graph reduction occurs in three phases : (a) Context determination (b) Local graph reduction and (c) Candidate pruning. Phases (a) and (b) operate on a neighborhood of a pair of vertices. The third phase however iterates over the graph and determines which vertex pairs to examine as we explain below.

3.2.1 Context Determination

We first determine a context (i.e. a set of vertices) for comparison. The basic idea of the algorithm is to define a context by using only *valuable* vertices in a given neighborhood. The inherent value is established in two ways : (a) the edge weight as compared to the maximum edge weight and (b) the contribution of the said vertex to the similarity / dissimilarity between the vertices being compared. We present Algorithm 1 that determines this context (or *"cross-enriched" neighborhood*).[2]

[2] getWJC() refers to weighted Jaccard coefficient

Algorithm 2 Local graph reduction

```
Input: The sets : $\mathrm{C}(i,j)$, $\mathrm{U}(i)$ and $\mathrm{U}(j)$
Output: MWE candidates $M$
Initialize $M \leftarrow \emptyset$
Let the set $\mathrm{U} \coloneqq \mathrm{U}(i) \cup \mathrm{U}(j)$, $|\mathrm{U}| = k$
Let $A = zeros(k,k)$
Construct adjacency matrix where $A(x,y) = p(\mathrm{U}_x, \mathrm{U}_y) + \sum_c^{\mathrm{C}(i,j)} p(c, \mathrm{U}_y)$
for All x,y within the same partition do
    if $A(x,y) \gg A(y,x)$ then
        Delete $\mathrm{U}_y$ locally
    else if $A(x,y) \approx A(y,x)$ then
        Add pair $< \mathrm{U}_x, \mathrm{U}_y >$ to $M$
    end if
end for
Return $M$
```

Algorithm 3 Generating MWE candidates

```
Input: A word graph $G = (V, E, W)$, cross-enrichment parameter $\eta$, word rarity parameter $\zeta$,
    co-occurrence parameter $\kappa$, positional index $idx$
Output: MWE candidates
Initialize $M_{op} \leftarrow \emptyset$
Let $V_d$ be the vertices $V$ sorted by descending order of degree
Initialize $\mathrm{M} \leftarrow \emptyset$
for $< V_i, V_j >$ in $V_d$ do
    $\mathrm{N}^e(V_i), \mathrm{N}^e(V_j) = crossEnrich(V_i, V_j, \eta)$
    compute $\mathrm{C}(i,j), \mathrm{U}_i, \mathrm{U}_j$
    $\mathrm{M} \leftarrow reduce(\mathrm{C}(i,j), \mathrm{U}_i, \mathrm{U}_j)$
end for
$\mathrm{M} \leftarrow filter(\mathrm{M}, \zeta, \kappa)$
for Group $g$ in $\mathrm{M}$ do
    $M_{op} \leftarrow expandPhrase(g, idx)$
end for
Return $M_{op}$, G
```

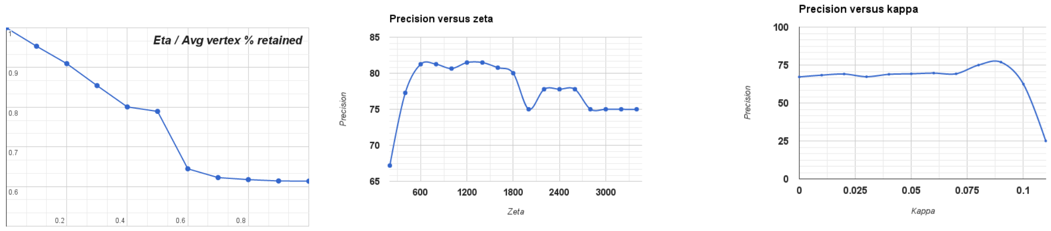
For a given vertices V_i and V_j, this algorithm effectively partitions their joint neighborhood into four disjoint sets:

1. Common vertices, $\mathrm{C}(i,j) \coloneqq \mathrm{N}^e(V_i) \cap \mathrm{N}^e(V_j)$
2. Uncommon vertices of i, $\mathrm{U}(i) \coloneqq \mathrm{N}^e(V_i) \setminus \mathrm{C}(i,j)$
3. Uncommon vertices of j, $\mathrm{U}(j) \coloneqq \mathrm{N}^e(V_j) \setminus \mathrm{C}(i,j)$
4. Ignored vertices, $\bigcup_k^{i,j} \mathrm{N}(V_k) \setminus \mathrm{N}^e(V_k)$

We only care about the common (C) and uncommon (U) vertices which act as inputs to the next phase.

3.2.2 Local Graph Reduction

In the next phase, we consider the sub-graph created by these three sets and perform local graph reductions as outlined in Algorithm 2. Essentially, we represent the local graph as a compressed adjacency matrix. For a given cell, $A(x,y)$, the weight in the matrix is set to the edge weight between vertices x and y plus the sum of weights from all common vertices to y. We then reduce the graph by either deleting



(a) Eta versus retained neighborhood

(b) Precision versus Zeta

(c) Precision versus kappa

Figure 2: Parameter tuning

a vertex if it is dominated by another node (i.e. this indicates that the dominated node never occurs independently) or by merging two vertices if their weights are equivalent (i.e. the given pair almost always co-occur - our original assumption). Note that these reductions are *local*.

3.2.3 Candidate Pruning & Phrase Expansion

In the final phase as illustrated in Algorithm 3, we output the final list of MWEs using a two step process. We first iterate over the graph that in turn calls Algorithm 1 and Algorithm 2. Next, we eliminate false positives based on two parameters : word rarity (ζ) and co-occurrence (κ). The former eliminates candidates that are composed of frequently occurring words, i.e. typical stopwords whilst the latter ensures a lower bound on the number of co-occurrences of the words that constitute the candidates. Finally, we use the positional index to reorder and expand the phrases as needed before outputting the final result.

A note about graph iteration is pertinent here. For a pair of vertices V_i and V_j input to the algorithms, it can be observed that the actual merge occurs on the vertices within the neighborhood of V_i, V_j and not on the vertices themselves. Thus, in order to cover as much graph as quickly as possible, the easiest strategy is to pick V_i, V_j in decreasing order of degree. Note that this also guarantees iteration in O(V) time.

Finally, although our method does seem similar to enumerating all n-grams and using some AM, we contend that this method can differentiate between nuances of usage due to the pairwise or cross-vertex iteration. In a typical n-gram approach, such contextual information is lost whereas in our method, it is equivalent to evaluating the n-grams in a limited context and is hence, more powerful. We now present details of parameter estimation and a short discussion on parameter sensitivity.

3.3 Parameter estimation

As we saw in Section 3.2, we use the following parameters:

- Enrichment parameter η : Determines which vertices in the current neighborhood will be considered
- Word rarity parameter ζ : Determines the level of rarity for a vertex to be considered
- Co-occurrence parameter κ : Determines the co-occurrence probability for an edge to be considered

We used the MH370 dataset (refer Table 2[3]) to find the optimum values of these parameters except for the *Enrichment parameter (η)* as described below. We obtained the value of η by evaluating the effect of varying η on a set of vertices and the neighboring vertices retained. We found a value of 0.6 to be a reasonable balance between over-pruning and retaining most vertices. Refer Figure 2 that demonstrates that a value of $\eta = 0.6$ does seem to have a large discriminatory power.

For estimating ζ and κ, we first used Algorithm 3 in a parameter-less mode (i.e. without filtering) and obtained all potential MWE candidates. For all such candidates, we established if the phrase indeed is a

[3] Collected when the MH370 flight had disappeared and investigation was underway

Sno	Dataset Name	# Tweets	Avg length	Vocabulary	Stopwords %	OOV (non English) %	Singleton %	Lang count
0	MH370	8,556	12.76	13,578	33.57	38.12	51.37	67
1	Brexit	18,488	10.96	23,734	30.74	40.40	34.9	58
2	DeleteYourAccount	2,244	6.85	2,445	44.45	24.39	42.33	45
3	Euro16	10,577	9.44	15,077	19.05	54.44	40.76	56
4	Giroud	6,697	9.35	8,576	17.91	58.10	45.98	51
5	PresidentObama	2,153	10.46	2,934	35.21	29.16	39.09	38
6	Pride	8,743	9.86	9,996	36.79	25.90	43.41	55
7	CalvinHarris	2,900	12.46	5,977	25.38	32.53	57.24	40
8	PokemonGO	7,019	10.81	15,068	32.14	44.22	64.32	60

Table 2: Dataset details

S.no	Phrase	DistScore	PhraseScore	HashtagScore	StopwordScore	Notes
1	clapback season	0.81	1	1	0	Ideal case : High scores for all three scores and no stopwords
2	Hillary Clinton	0.48	1	1	0	Named Entity but tokens can appear far apart
3	delete emails	0.37	1	0.04	0	Phrase query alone can be misleading
4	right now	0.45	1	1	1	Other measures compensate for lack of high distance score
5	if you	0.7112	1	1	2	High scores do not always mean MWEs

Table 3: Examples of need for four features

MWE using the Microsoft Web Language Model API [4] and the PMI metric [5]. We then measured system precision by varying each of the parameters independently as shown in Figure 2. We found the optimal values to be $\zeta = 1000$ and $\kappa = 0.01$.

4 Data and Experiments

Dataset	Dice	PMI	LogL	TwoT	T-Score	GRePE	# Candidates	Actual MWEs
Brexit	37.87	18.60	43.69	15.14	24.76	**62.40**	737	193
DeleteYourAccount	50.38	42.65	42.04	30.44	36.70	**66.14**	110	47
Euro16	32.48	14.40	**63.42**	39.28	61.17	42.36	328	67
Giroud	21.69	9.75	**78.38**	47.94	78.25	50.65	62	29
PresidentObama	**95.58**	86.26	92.72	2.04	92.72	59.93	36	15
Pride	54.37	40.18	**58.34**	15.30	39.41	51.55	137	56
CalvinHarris	**89.09**	41.31	66.52	73.62	77.58	84.17	33	21
PokemonGO	28.95	34.63	19.04	8.34	15.24	**58.46**	85	25
MAP / Total	51.30	35.97	58.02	29.01	53.23	**59.46**	791	453

Table 4: Experimental results

4.1 Datasets and data collection

As outlined in Section 1, our primary focus lies in extracting and analyzing MWEs from short text documents, namely tweets. Given the diverse nature of users, languages employed and topics discussed on Twitter[6], we wanted to achieve as broad coverage as possible. For over two weeks[7], we collected tweets for selected trending topics at different times of day. The choice of the selected topics was based on volumes as reported by Twitter plus the perceived global reach of the topic itself. However, for the final analysis we only used a subset of our crawled data as any sets with less than 2000 unique tweets were discarded. Although Twitter provides its own language identification, we used *langid* (Lui and Baldwin, 2012) for our use to allow generalization to other data sources (like Facebook) later.

A summary of the datasets is provided in Table 2 that captures the language and vocabulary spread for each hashtag. Note that the volume of tweets notwithstanding, each HashTag has tweets in at least 30 different languages, the average tweet length is only about 10 words and the word frequency distribution has a significant long-tail with about 40% of the words occuring just once.

4.2 Automatic Evaluation

As outlined in Section 2, we evaluate our system on precision as against recall and compare the system generated MWEs with those generated by standard AMs. Since we are computing Average Precision,

[4] https://www.microsoft.com/cognitive-services/en-us/web-language-model-api

[5] We tested different AMs and found PMI to be the most effective in this scenario

[6] https://about.twitter.com/company

[7] Roughly June 21 2016 - July 10 2016

the metric value is sensitive to the size of the result set considered. Given that our system produces limited number of MWEs, we restrict the output of compared AMs to be equal to the number of MWEs generated by our system.

In order to ascertain if a generated phrase is indeed a MWE, we performed two levels of evaluation. At the first level we use the Twitter Search API[8] as follows. For every candidate $W = (w_1, w_2)$, we execute three queries while restricting each query to top 25 unique results [9] : (a) w_1 w_2 (which is equivalent to w_1 AND w_2) (b) the phrase "w_1 w_2" (c) concatenation w_1w_2. Each result set is then converted to a corresponding numeric score as below

1. DistanceScore = Average normalized token distance between tokens w_1 and w_2
2. PhraseScore = Number of returned results / 25
3. HashtagScore = Number of returned results / 25

While the latter two scores approximate the probability of the phrase occurring either as separate words or concatenated together, the first score is a proxy for how frequently do the constituent words appear next to each other (as in a phrase) versus co-occurring in a tweet. Additionally, we add a fourth parameter, an integer stopword score $[0, 2]$ that acts as a regularization parameter to penalize phrases that contain stopwords which are bound to return a large number of results. We present some examples to illustrate the need for all four values in Table 3. We trained a simple multinomial logistic regression classifier on the MH370 dataset on manually evaluated MWE candidates with a 70% true label precision.

As second layer of screening, we assign one of the 15 POS labels as listed in Table 1 and double check that the extracted candidates are in fact MWEs. Note that we translate phrases from languages other than English into English before assigning the POS tags. We admit this is slightly lossy but we view at as a way to project all MWEs in the same token space for simplicity. Thus, for each dataset, we compute the Average Precision by using the true class labels obtained as explained above. We present the results in Table 4 along with Mean Average Precision (MAP).

We additionally compare the overlap between our method and the different AMs in Table 5a as well as splits by POS tag type in Table 5b. These tables show that although the different AMs do not necessarily generate the same candidate list (except LogLikelihood and T-score), the comparable POS split percentages indicate inherent bias within the dataset.

4.3 Discussion of results

We must take a moment to explain and examine the results. Although, it may not seem that our method is a vast improvement over other AMs when looking at the MAP, it must be noted that we do not produce "ranked" results as such and only candidates. We used a fixed ordering based upon the co-occurrence probability of phrases and a better ranking mechanism may exist but was not explored. The performance of the AMs is also bound to suffer when the full result sets are used. Further, except for the PresidentObama dataset, our method places within top 3 where it is not the best performing method. Comparing against Table 2, the method seems to suffer for predominantly English datasets (low OOV% - Pride, PresidentObama etc) but better for multilingual datasets (Brexit, PokemonGO). Thus, we could in principle augment our method with either AMs or existing POS based approaches for English to further improve performance. However, it can be concluded that overall the method returns a small and fairly precise set of MWEs as compared to AMs and enumerating all bigrams.

5 Future Work and Conclusions

In summary, we can enumerate our contributions as (a) we presented a language agnostic method for extracting MWEs from Twitter (b) we explored the performance of different AMs in a similar setting and (c) we showed a method for automatic evaluation of extracted MWEs. As an extension to this work, we would like to further analyze our results and study the effect of Twitter and social media specific features

[8] https://dev.twitter.com/rest/public/search

[9] With a page size of 10 tweets, this seemed a good choice for tweet depth without running too many queries

Measure	**Dice**	**PMI**	**LogL**	**TwoT**	**T-Score**	**GRePE**
Dice	NA	0.00	0.00	0.00	0.00	2.23
PMI	0.00	NA	0.00	0.00	0.00	0.00
LogL	0.00	0.00	NA	0.00	59.53	3.00
TwoT	0.00	0.00	0.00	NA	5.68	0.00
T-Score	0.00	0.00	59.53	0.00	NA	1.67
CDAM	2.23	0.00	3.00	0.00	1.67	NA

(a) AM Overlap

Measure	ADJP	NP	ADVP	VP
Dice	2.81	85.92	1.41	9.86
PMI	5.77	73.08	0.00	21.15
LogL	1.11	75.82	0.00	25.93
TwoT	0.05	52.50	0.00	42.50
T-Score	1.45	72.46	0.00	26.09
GRePE	1.64	83.61	0.00	14.75
Avg	**2.96**	**73.90**	**0.23**	**22.91**

(b) Split by POS tags

Table 5: Comparison between AMs

on MWE usage. Namely does internet language, hashtags and code switching impact how MWEs are used? We would also like to explore if the extracted MWEs can be utilized for other downstream tasks like generating summaries or automatic bilingual tweet alignment. We believe such work would help in developing resources for resource poor languages as well as aid in better understanding and modeling language usage on social media.

References

Muhammad Abdul-Mageed, Sandra Kübler, and Mona Diab. 2012. Samar: A system for subjectivity and sentiment analysis of arabic social media. In *Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis*, pages 19–28. Association for Computational Linguistics.

Mirna Adriani and CJ Van Rijsbergen. 2000. Phrase identification in cross-language information retrieval. In *Content-Based Multimedia Information Access-Volume 1*, pages 520–528. LE CENTRE DE HAUTES ETUDES INTERNATIONALES D'INFORMATIQUE DOCUMENTAIRE.

Tetske Avontuur, Iris Balemans, Laura Elshof, Nanne Van Noord, and Menno Van Zaanen. 2012. Developing a part-of-speech tagger for dutch tweets. *Computational Linguistics in the Netherlands Journal*, 2:34–51.

Shane Bergsma, Paul McNamee, Mossaab Bagdouri, Clayton Fink, and Theresa Wilson. 2012. Language identification for creating language-specific twitter collections. In *Proceedings of the second workshop on language in social media*, pages 65–74. Association for Computational Linguistics.

Marine Carpuat and Mona Diab. 2010. Task-based evaluation of multiword expressions: a pilot study in statistical machine translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 242–245. Association for Computational Linguistics.

Joaquim Ferreira Da Silva, Gaël Dias, Sylvie Guilloré, and José Gabriel Pereira Lopes. 1999. Using localmaxs algorithm for the extraction of contiguous and non-contiguous multiword lexical units. In *Portuguese Conference on Artificial Intelligence*, pages 113–132. Springer.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, pages 198–206.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Spence Green, Marie-Catherine De Marneffe, John Bauer, and Christopher D Manning. 2011. Multiword expression identification with tree substitution grammars: A parsing tour de force with french. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 725–735. Association for Computational Linguistics.

Mengdie Hu, Shixia Liu, Furu Wei, Yingcai Wu, John Stasko, and Kwan-Liu Ma. 2012. Breaking news on twitter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2751–2754. ACM.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A Smith. 2014. A dependency parser for tweets.

Anoop Kunchukuttan and Om Prakash Damani. 2008. A system for compound noun multiword expression extraction for hindi. In *6th International. Conference on Natural Language Processing*, pages 20–29.

Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. 2010. What is twitter, a social network or a news media? In *Proceedings of the 19th international conference on World wide web*, pages 591–600. ACM.

Nikhil Londhe, Vishrawas Gopalakrishnan, Aidong Zhang, Hung Q Ngo, and Rohini Srihari. 2014. Matching titles with cross title web-search enrichment and community detection. *Proceedings of the VLDB Endowment*, 7(12):1167–1178.

Marco Lui and Timothy Baldwin. 2012. langid. py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 system demonstrations*, pages 25–30. Association for Computational Linguistics.

Antonio Moreno-Ortiz, Chantal Pérez-Hernández, M Ángeles Del-Olmo, et al. 2013. Managing multiword expressions in a lexicon-based sentiment analysis system for spanish. *NAACL HLT 2013*, 13:1.

Joakim Nivre and Jens Nilsson. 2004. Multiword units in syntactic parsing. *Proceedings of Methodologies and Evaluation of Multiword Units in Real-World Applications (MEMURA).*

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. Association for Computational Linguistics.

Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, pages 1320–1326.

Darren Pearce. 2002. A comparative evaluation of collocation extraction techniques. In *LREC*.

Ted Pedersen, Satanjeev Banerjee, Bridget T McInnes, Saiyam Kohli, Mahesh Joshi, and Ying Liu. 2011. The ngram statistics package (text:: nsp): A flexible tool for identifying ngrams, collocations, and word associations. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 131–133. Association for Computational Linguistics.

NH Rais, MT Abdullah, and RA Kadir. 2011. Multiword phrases indexing for malay-english cross-language information retrieval. *Information Technology Journal*, 10(8):1554–1562.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. Multiword expressions in the wild?: the mwetoolkit comes in handy. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*, pages 57–60. Association for Computational Linguistics.

Carlos Ramisch, Vitor De Araujo, and Aline Villavicencio. 2012. A broad evaluation of techniques for automatic acquisition of multiword expressions. In *Proceedings of ACL 2012 Student Research Workshop*, pages 1–6. Association for Computational Linguistics.

Ines Rehbein. 2013. Fine-grained pos tagging of german tweets. In *Language Processing and Knowledge in the Web*, pages 162–175. Springer.

Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. 2009. Improving statistical machine translation using domain bilingual multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 47–54. Association for Computational Linguistics.

Alan Ritter, Oren Etzioni, Sam Clark, et al. 2012. Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Computational Linguistics and Intelligent Text Processing*, pages 1–15. Springer.

Nathan Schneider, Spencer Onuffer, Nora Kazour, Emily Danchik, Michael T Mordowanec, Henrietta Conrad, and Noah A Smith. 2014. Comprehensive annotation of multiword expressions in a social web corpus.

Yutaro Shigeto, Ai Azuma, Sorami Hisamoto, Shuhei Kondo, Tomoya Kouse, Keisuke Sakaguchi, Akifumi Yoshimoto, Frances Yung, and Yuji Matsumoto. 2013. Construction of english mwe dictionary and its application to pos tagging. In *Proc. of the 9th Workshop on Multiword Expressions*, pages 139–144.

R Mahesh K Sinha. 2011. Stepwise mining of multi-word expressions in hindi. In *Proceedings of the Workshop on Multiword Expressions: from Parsing and Generation to the Real World*, pages 110–115. Association for Computational Linguistics.

Thamar Solorio and Yang Liu. 2008. Part-of-speech tagging for english-spanish code-switched text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1051–1060. Association for Computational Linguistics.

Thamar Solorio, Elizabeth Blair, Suraj Maharjan, Steven Bethard, Mona Diab, Mahmoud Gohneim, Abdelati Hawwari, Fahad AlGhamdi, Julia Hirschberg, Alison Chang, et al. 2014. Overview for the first shared task on language identification in code-switched data. In *Proceedings of The First Workshop on Computational Approaches to Code Switching*, pages 62–72. Citeseer.

Yulia Tsvetkov and Shuly Wintner. 2014. Identification of multiword expressions by combining multiple linguistic information sources. *Computational Linguistics*, 40(2):449–468.

Tim Van de Cruys and Begona Villada Moirón. 2007. Semantics-based multiword expression extraction. In *Proceedings of the Workshop on A Broader Perspective on Multiword Expressions*, pages 25–32. Association for Computational Linguistics.

Yogarshi Vyas, Spandana Gella, Jatin Sharma, Kalika Bali, and Monojit Choudhury. 2014. Pos tagging of english-hindi code-mixed social media content. In *EMNLP*, volume 14, pages 974–979.