Institut für maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Corpus-Driven Study of Context-Dependent Behavior of Idiomatic MWEs and their Constituents: An Implementation Perspective

Tarek Mehrez & Anna Konobelkina
Project Paper

Supervisor: Dr. phil. Cerstin Mahlow

Submission Date March 24, 2015

Contents

1 Introduction **2**

2 Theoretical & Linguistic Background **2**

2.1 Definition of MWEs 2

2.2 Arguments Supporting Context-Defined MWEs 3

3 Method **4**

3.1 Definitions & Initial Tools 4

3.2 Directory Structure 5

3.3 Corpus 5

3.4 Implementation 6

3.4.1 MWE Annotation 7

3.4.2 Compile IOB File 7

3.4.3 Reformatting text 8

3.4.4 Constructing phrases 8

3.4.5 Measuring Distance 8

3.4.6 The Bigger Picture 8

3.5 Measuring Contexts 9

3.5.1 Baseline Calculation 10

3.5.2 Vector Extraction 11

4 Results & Discussion **11**

4.1 Results 11

4.2 Discussion & Limitations 12

4.3 A Deeper Look into Clustering 12

5 Conclusion **12**

1 Introduction

In this work, we investigate the definition of multi-word expressions (MWEs). We tried to introduce a scientifically proven method to support our hypothesis in how we can define an MWE. Our hypothesis is concerned with contexts in which words may appear. Specifically, the contexts in which an MWE and its constituents may appear. By definition, those contexts should be different for the MWE to have its special meaning, that made it necessary to use in the first place. Linguistically, definitions may differ. Researchers took different paths towards describing MWEs concretely. Therefore, we decided to focus on a certain subclass of MWEs that supports our hypothesis.
To that end, we implemented a tool that measures contexts and their differences when it comes to an MWE and its constituents. We made use of word-learned vectors to act as the comparison metric.
This work is counted as a pre-requisite for the "Detecting and classifying multi-word expressions" course, and not as a novel endeavour or a scientific proposal to prove. However, that doesn't deny the fact that further discussions may emerge from the results we introduce in this paper, hoping that we can contribute to a more concrete definition of MWEs.

Section 2 contains a brief linguistic background on the general definitions of MWEs, but an extensive linguistic description of this work is described in its twin document [**?**]. The details of the implementation are explained in section 3. Consequently, the results and conclusion are discussed in sections 4 and 5 respectively.

2 Theoretical & Linguistic Background

2.1 Definition of MWEs

This work targets the definition of an MWE. According to standard explanations, an MWE has different ways to be defined in terms of form an meaning. As we stick to the literal meaning, it would obviously mean an expression with multiple words or lexemes. What these words represent or have in common is something that is not agreed on upon different researchers. According to [1], an MWE could be easily identified as an expression whose words are separated by white spaces. However, they added a counter argument that disproves the whitespaces approach: other languages such as German has concatenated compound nouns that could be still defined as MWEs such as *Kontaktlinse* "contact lens".

A definition which is somehow straight forward is provided by [12], they argued that an MWE, as stated in above, is a group of lexical items that consists of multiple constituents that represent some idiomaticity. We believe this is where different arguments may appear. Idiomaticity could be defined lexically, syntactically, semantically, pragmatically or statistically. This makes things a bit vague as a concrete definition is still missing.

MWEs such as "living room" refer to a certain entity and meet the first requirement by having multiple constituents. But whether the literal meaning is different or not, this is where the argument may emerge.

Different classes of MWE should be taken into consideration as well. For example, named entities are a class of tokens that could intersect with the set of MWEs. By intuition not all named entities are MWEs, and vice versa. The named entity "San Fransisco", or "The United Nations" are definitely MWEs, whereas "England" is obviously not.

Although some references may consider "England" as an MWE [16], in this work, this is simply ignored as it contradicts the aforementioned definition.

Generally, one of the directions that have been taken is the difference in contexts. For example, idiomatic MWEs such as "cross the line" have interesting characteristics that should be taken into consideration [12] Given that in the mentioned example the literal meaning is not the intended meaning, the expression's constituents may appear in different contexts separately than as a whole [4]. And this is the main point of our focus. This means that "cross the line" in a debating context will be different than "cross the line" in a sports context. Same for the constituents "cross", "the" and "line" appearing in different contexts separately.

Other classes of MWEs such as compound nouns might also be interesting. The MWE "world cup qualifying matches" has a lot to say when its contexts is compared to its constituents, as we did previously. This MWE could refer to different contexts such as sports or politics.

This of course doesn't apply to all types of MWEs. For example particle verbs won't be really interesting in that case [8]. The verb "find out" will not necessarily have different contexts when we break it down. What brings us to the question whether we can consider particle verbs as MWEs or we can basically ignore context as a main factor of defining MWEs when it comes to some subclasses.

Nevertheless, the aforementioned argument can still prove that one of the main necessities to define MWEs, or at least a certain subclass, would be defining the contexts in which they appear, separately and as a whole.

For that reason the main goal behind this work is to investigate the difference in contexts between the MWE as a whole trying to reach a better understanding of what an MWE is and how to define it.

2.2 Arguments Supporting Context-Defined MWEs

This could be interesting if we formulated this argument in a top-down approach. To be more specific, how things will differ if we decided to decompose the MWE. If there no big difference after the decomposition, then it wasn't really necessary to use in the first place. That doesn't mean that MWEs such as "living room" should be deprecated, but rather labelling it as an MWE should be reconsidered.

From a practical perspective, less attention should be paid to the linguistics details, if we are dealing with a bigger NLP target or task. However, since in that case we are more into the linguistic definition of MWEs, we will do it the other way around trying to prove it from a practical perspective.

The reason why we are concerned with this is how problematic is it to deal with MWEs in general NLP applications. Needless to mention the titles "Multi-word expressions: A pain in the neck for NLP" in [12], or "Introduction to the special issue on multiword expressions: Having a crack at a hard nut" [15].

In general NLP tasks, people can just ignore MWEs and deal with them as ngrams as in [13]. That way MWEs with more words than the ngrams are ignored, and of course all single phrases will be considered as MWEs. Others tried to identify MWEs to be able to proceed with other tasks such as machine translation and text alignment in parallel corpora [10]. That proves an emerging necessity in dealing with MWEs, and defining them concretely. More about the assumptions, NLP steps, and implementing the necessary tools will be described in the upcoming section.

3 Method

This section explains the technical details of the method used to extract MWEs, identify their contexts and measure discrepancies. To that end a Java tool was implemented. The tool handled pre-processing as well as the contexts measurements. Further details are explained later in this section. As mentioned before, the twin document [6] of this work contains a more linguistic perspective on what has been done.

3.1 Definitions & Initial Tools

The technical implementation that can prove our hypothesis was a bit tricky. A concrete definition of a context was the hardest. Different works have tackled the idea of a context, and how can we divide a piece of text into different contexts or clusters according to word co-occurrences. Several methods for topic extraction and tracking have been introduced. We considered the common ones starting with the basic models such as TFIDF, co-occurrence matrices, latent semantic analysis or basic text cluastering [2, 7], or a slightly more complex methods such as latent dirichlet allocation [3].

But since none of those was straightforward to use for our hypothesis, we had to tailor another approach around our special case. Therefore, we decided to use a model which is a mixture between text clustering and learning word vectors, to have both as our technical mapping of a context. The basic idea is to measure distance between learned vectors, then check to which clusters did the MWE and the constituents belong. That way we have an idea of the contexts in which they appear.

Vectors were produced using context-window methods such as skip-gram and continuous bag of words (cbow), as well as a mixture between context-window methods and matrix factorization methods called glove. Then clustering was performed on top of the produced vectors using k-means algorithm. The notion of word-learned vectors with MWEs is not really novel. To our knowledge, it has been applied before in other works such as [5, 14]. More details will be explained in the next subsections.

Two external open-source tools were needed to aid the vector calculation and clustering processes. We used the tools word2vec and glove to produce the vectors for the lexicon. Clusters were produced on top of the vectors using word2vec.

3.2 Directory Structure

The directory structure is divided as follows

- src: the java source code
- libs: external tools for the java application
- corpus: the wiki50 corpus
- doc: documentation
- tools: external tools (word2vec & glove)
- vectors: the produced vectors and classes/clusters
- results: results files

A snapshot of the directory structure is demonstrated in figure 1

3.3 Corpus

To prove our hypothesis, we needed a corpus that already provided manual annotations of MWEs to avoid the error propagated by automatically identifying them and of course, save time and effort. Therefore, we decided to proceed with the wiki50 ccorpus [16].

The corpus consists of 50 scientific articles as text files and the respective MWE annotations alongside their types. Types for MWEs included among others `MWE_COMPOUND_NOUN` for nominal compounds, `MWE_COMPOUND_ADJ` for adjectival compounds, `MWE_OTHER` for foreign phrases and other MWEs. MWEs were annotated using the starting and ending character indices in the corresponding text file. Also a compiled list of all MWEs in the corpus was available.

We combined all text files in one to apply the vector extraction and clustering. The training file had about than 95K tokens, with a distinct vocabulary size of about 15K tokens. A huge drawback was the lack data. The number of tokens was not enough to train a large-scale vector space model. But we also had limited options of corpora with well-annotated MWEs, so we decided to stick to the wiki50 corpus.

Limitations of the corpus and possible enhancements are discussed in section 4.

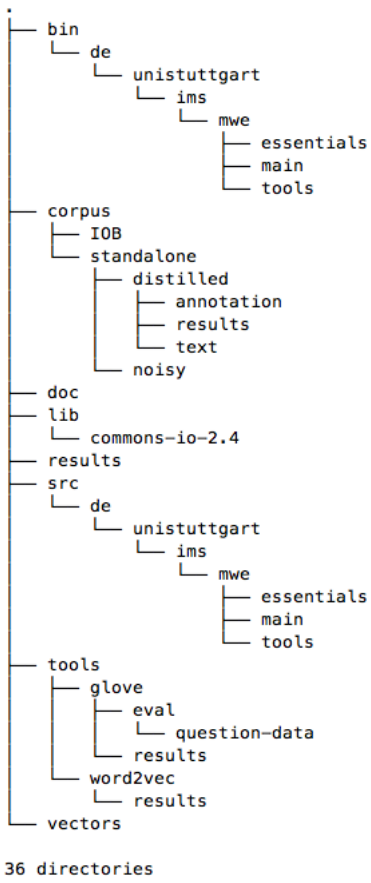

Figure 1: Snapshot of the root directory

3.4 Implementation

The implemented tool is a combination of several steps. With the help of the vector-calculation tools word2vec and glove, we were able to construct a pipeline where all intermediate steps could be treated as a black box. That was done on purpose so that any of the steps could be used separately as an independent tool. The implementation could be accessed via the following link. `https://github.com/tarekmehrez/MWE-in-Context`.

The tool that handled the main traffic was implemented in Java and was combined with the external open-source tools to produce the final results. To run the tool, one can pass any of the possible parameters to that jar file `mwe.jar`. Possible parameters and their respective usage are described in detail in the upcoming subsections.

3.4.1 MWE Annotation

The wiki50 MWE annotations were marked with character indices. In order to produce a list of MWEs, rather than just indices, we had to include this into the implementation. The first option `--annotate-corpus` takes the root directory to text and annotation files, to produce the actual MWE list for each text file.

As we implemented that option, we noticed that the indices were corrupt, basically some of the MWEs were extracted successfully with the character indices, others had missing characters or extra ones. That wasn't consistent across all files, which means that the character-based annotations wasn't really accurate in the corpus.

Luckily enough, the corpus had another compiled list of all phrases in all files including MWEs. So instead of this option, we used the compiled list. But this option was kept in case we needed separate annotations for each file.

3.4.2 Compile IOB File

The compiled list had all phrases in all text files, each in a separate line. MWEs were marked though, so basically we could access this list, extract MWEs with their types, and write them in a final list containing only MWEs.
The parameter `--compile-iob` was implemented and could be used by running the jar file, with the path to the list provided by the corpus (the IOB file). This parameter produces a list of all MWEs and their types separated by commas. The entries of this file were pre-processed by removing special characters and lowercasing the MWEs to match the text pre-processing described later. This list helped producing specific statistics about the results for each separate type. A snapshot of the compiled list is represented in the figure 2.

```
military air traffic control,MWE_COMPOUND_NOUN
free falling seal team,MWE_COMPOUND_ADJ
flying aircraft,MWE_COMPOUND_NOUN
military aircraft,MWE_COMPOUND_NOUN
touch and go approach,MWE_COMPOUND_NOUN
combat gear,MWE_COMPOUND_NOUN
makes a quick decision,MWE_LVC
emergency cord,MWE_COMPOUND_NOUN
reserve chute,MWE_COMPOUND_NOUN
roman catholic,MWE_COMPOUND_ADJ
```

Figure 2: Snapshot of the compiled MWEs list

3.4.3 Reformatting text

As we treated the whole corpus as one big text file containing all articles, we had to pre-process this file to make sure it's suitable for further manipulations. The option `--reformat-text` was implemented to take the text file, lowercase it, remove special characters and write the results into a new file. That was necessary to make sure that no special characters can intrude the extraction of MWEs or matching them for further steps such as vector extraction.

3.4.4 Constructing phrases

In order to produce vectors for the whole corpus, we had to treat each MWE as one token or entry, so that we can have a single vector for this exact MWE, to be compared with its constituents later on. Therefore, we had to implement an option that takes both the text file and the compiled MWEs list, and returns back the text file with marked MWEs. The parameter `--construct-phrases` produces the same exact text file entered, but this time with the MWEs marked by underscores between its tokens. So for example, the term "world cup" appearing in the initial text file will appear as "world_cup" in the resulting file.

3.4.5 Measuring Distance

Finally, we reach the most important module, the one that measures distances and calculates the final results. The parameter `--compute-distances` takes the file containing the vectors, the compiled MWEs list and the clusters file, to produce the final results.

The results file contains all MWEs, their types, the euclidean distance between them, and their constituents, and the baseline for each MWE. Finally, it contains the results for each MWE type including the number of correctly classified instances, or those which had different contexts than their constituents and the final accuracy.

Figure 3 shows a snapshot of one of the results files.

Vector extraction methods and baseline calculations are described in the upcoming subsections.

3.4.6 The Bigger Picture

Now that we had all pieces, we can just formulate the whole procedure by defining one big pipeline of steps. Example commands on how to run each step are available in the running script `run.sh`.

1. All text files are compiled in one and processed as described above
2. All MWEs are gathered in one list with their types after being processed

```
mwe,type,distance,baseline
gained_international_attention,gained_international_attention,MWE_LVC,1.1784566162370547,4.351922229059476
military_academy,military_academy,MWE_COMPOUND_NOUN,1.772206392312278,4.8219982195036115
adult_education_classes,adult_education_classes,MWE_COMPOUND_NOUN,1.4376989571042174,5.21016089452562
social_outcast,social_outcast,MWE_COMPOUND_NOUN,0.8327182981696151,8.233731883128815
housewarming_party,housewarming_party,MWE_COMPOUND_NOUN,1.4106626763318542,5.1096660831737335
draws_his_main_support,draws_his_main_support,MWE_LVC,2.6497980233710856,5.109857980523138
.
.
.
################################
Number of MWEs: 2705
Accuracy for each MWE type:
MWE_COMPOUND_ADJ: 34 out of 76 with 44.74\% accuracy
MWE_IDIOM: 17 out of 18 with 94.44\% accuracy
MWE_OTHER: 13 out of 21 with 61.9\% accuracy
MWE_LVC: 224 out of 361 with 62.05\% accuracy
MWE_COMPOUND_NOUN: 1810 out of 2848 with 63.55\% accuracy
Total accuracy of difference in contexts: 79.19\%
```

Figure 3: Snapshot of the results file

3. MWEs are matched in the text files to produce vectors for MWEs as a whole
4. K-means implementation in word2vec is applied to cluster the vocabulary.
5. Vectors are extracted using word2vec and glove
6. Vectors, clusters and MWEs list are fed back to that tool to measure discrepancies and calculate results

Figure 4 describes this implemented pipeline.

3.5 Measuring Contexts

As discussed before, contexts were basically clusters calculated according to context-windows. And in order to measure differences between clusters, we had to calculate euclidean distances between vectors.

So, for each MWE to pass the test, it must meet the following criteria:

1. The distance between the MWE's vector and the average vector of its constituents must be greater than its baseline
2. The MWE's vector must be different from all vectors of its constituents

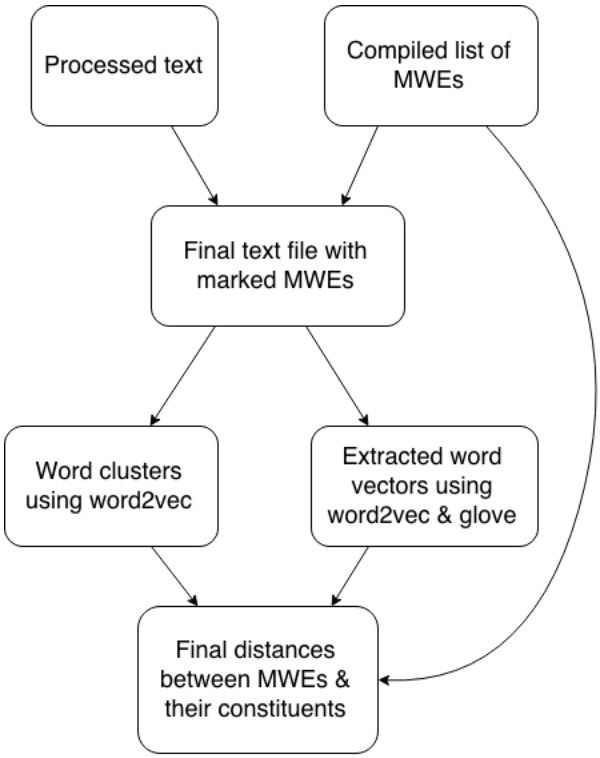

Figure 4: Snapshot of the compiled MWEs list

3. The MWE must lie within a different cluster than its constituents

Then and only then, an MWE will pass the criteria and is said to have different context than its constituents.

3.5.1 Baseline Calculation

Calculating the baseline was a bit tricky, since we needed an evaluation metric for our claim to make sense and prove our hypothesis. Considering that we have word vectors as our main method of calculating contexts, we calculated our baseline to be the maximum distance between the MWE and all other phrases occurring in the same cluster. That way we ensure that this distance represents the edge of the cluster and that any greater distance should naturally occur out of the cluster and therefore, be in another context. For that purpose, we calculated for each MWE its own baseline to be compared with the average distance of its constituents as explained before.

3.5.2 Vector Extraction

In order to extract vectors, we used the open-source tools word2vec and glove as mentioned before. The main difference is the implementation of both. Glove extracts the vectors by following a global training approach. Basically a co-occurrence matrix is constructed following any global matrix factorization method (e.g, Latent Semantic Analysis), then further training is done globally rather for a certain context, trying to minimize the error between the word (co-occurrence) vectors and the probability of their co-occurrences [11]. This error function is basically the mean squared error cost function. This global training considers the whole data set in each iteration rather than just a certain window.

On the other hand, context window methods that are implemented in word2vec, focuses on a certain context instead of the whole data set. Examples of context window models that are implemented in word2vec are the continuous bag of words models (CBOW) and the skip-gram model, both are described in [9].

All methods were used to compare different results in our task and surprisingly, they all produced the same exact results.

4 Results & Discussion

4.1 Results

The results were pretty promising as they proved our hypothesis, at least in our controlled environment.

We tested the system with all types of vectors (cbow, skip-gram & glove) and different number of clusters (10, 50, 100, 250, 500).

The results are shown in the following table.

Clusters	Accuracy
10	79.19%
50	95.19%
100	97.82%
250	98.19%
500	99.3%

Table 1: Results with different number of clusters

Obviously, as the number of clusters increases, hence the number of contexts, we succeed in distinguishing between more MWEs and their constituents. The significant increase between 10 clusters and 50 clusters has no concrete explanations though, as the increase from 50 and 500 is not as big. Nevertheless, this proves that clusters resembling the notion of contexts was more or less successful.

4.2 Discussion & Limitations

As the results look a bit optimistic, we may assume that a lot of factors may affect the reliability of this approach. First of all, the corpus didn't have enough data as we mentioned before. But again as we were constrained with an annotated corpus, wiki50 seemed like a good candidate.

Concerning the method, the mentioned criteria did a lot of assumptions on how vectors and clusters represent contexts, but that's not necessarily true as a concrete relation between vectors and clusters wasn't really there. Using different clustering and topic identification algorithm would have been beneficial to compare results.

As for using different types of vector extraction tools that turned out to be unnecessary as the results were surprisingly identical for cbow, skip-gram, and glove given the differences between their implementations. A possible reason would be our metric of measuring contexts. A context measurement metric other than euclidean distances could yield different results for different vector-learning implementations.

One major drawback was the huge loss in the number of MWEs from the original compiled list and the produced vectors. Initially, the number of MWEs extracted from the corpus was 3324, but the vector-extraction tools produced vectors for just 2705. That loss in the number of MWEs could have made a difference in the results.

Likewise, further enhancements could be further NLP pre-processing, including stop words removal (except those which appear in MWEs) and some stemming to decrease the number of vectors for similar words.

4.3 A Deeper Look into Clustering

Concerning the implementation of the k-means algorithm in word2vec, we did further investigation on how reasonable were the extracted contexts. To that end, some of the MWEs were selected to visualize their extracted contexts. Figures 5 and 6 show context differences between the MWEs "away_fans" and "taking_charge" and their constituents with total number clusters equal to 500.

Words in the previous figures are manually selected, to give a glance of the word distributions within the clusters. As shown, words within the same cluster don't have one concrete meaning or context, given that the number of clusters is 500. That's why some of the co-occurring words had slightly different relevance or no relevance at all.

Nevertheless, those figures still prove the hypothesis given that the contexts don't really overlap.

5 Conclusion

As a more concrete definition of multi-word expressions (MWEs) is needed, we presented in this paper a hypothesis that may help reach this definition. This contribution could be regarded for both linguistic and

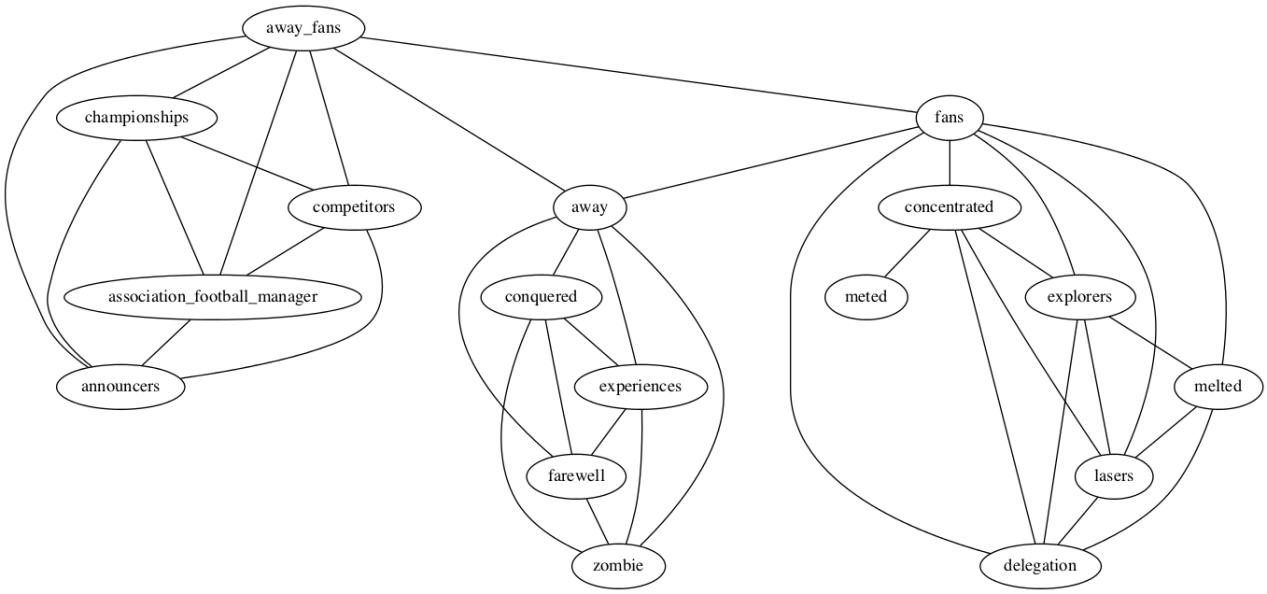

Figure 5: Distribution of words for the MWE away_fans

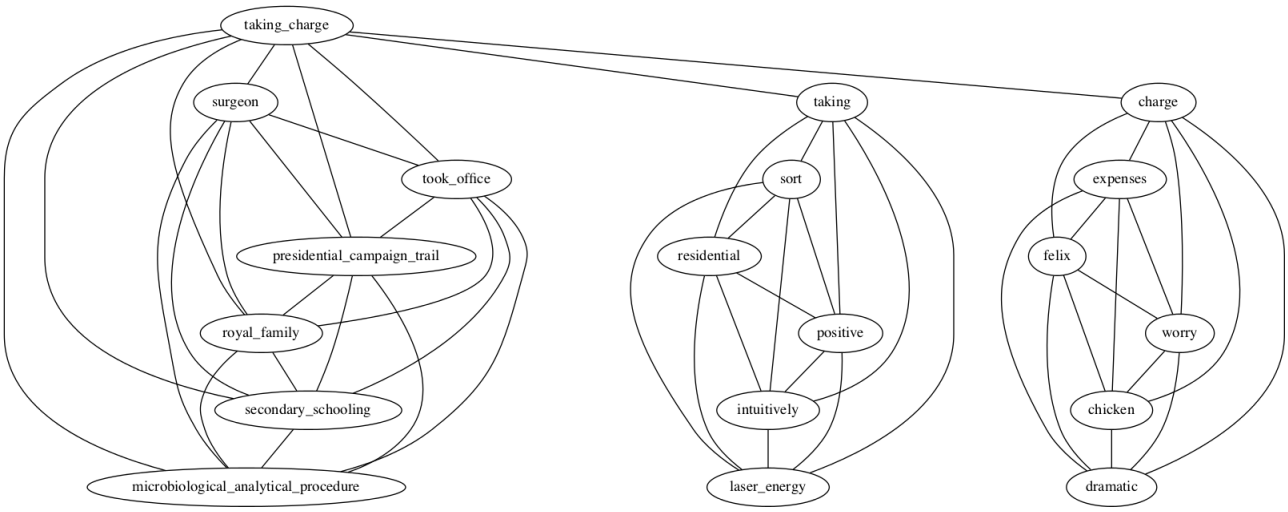

Figure 6: Distribution of words for the MWE taking_charge

technical NLP purposes. Our hypothesis claims that a certain subclass of multi-word expressions appears in different contexts than their constituents. That was motivated by our intuition that for a MWE to be useful, and for it to be necessary to use, it must have a different intended meaning than the literal meaning of its constituents.

Since different sub classes of MWEs are always reconsidered whether they should be labelled as MWEs or

not, this work doesn't cover all MWEs, but rather those that support the idea of context differences (e.g, idiomatic MWEs).

In order to prove a linguistic definition from an practical perspective, we implemented a tool that measures contexts between MWEs and their constituents. Both word-learned vectors and word clusters were used to identify a context.

Our results showed that there is a lot of potential behind this idea, given the non-primitive criteria and the visualized difference in contexts. Yet, a lot of enhancements could be taken into consideration to produce better and more reliable results.

References

[1] Timothy Baldwin and Su Nam Kim. Multiword expressions. *Handbook of Natural Language Processing, second edition. Morgan and Claypool*, 2010.

[2] Florian Beil, Martin Ester, and Xiaowei Xu. Frequent term-based text clustering. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–442. ACM, 2002.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] Nicoletta Calzolari, Charles J Fillmore, Ralph Grishman, Nancy Ide, Alessandro Lenci, Catherine MacLeod, and Antonio Zampolli. Towards best practice for multiword expressions in computational lexicons. In *LREC*, 2002.

[5] Graham Katz and Eugenie Giesbrecht. Automatic identification of non-compositional multi-word expressions using latent semantic analysis. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 12–19. Association for Computational Linguistics, 2006.

[6] Anna Konobelkina and Tarek Mehrez. Corpus-driven study of context-dependent behavior of idiomatic mwes and their constituents: A linguistic perspective. Technical report, Institute for Natural Language Processing, University of Stuttgart.

[7] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[8] Francesca Masini. Multi-word expressions between syntax and the lexicon: the case of italian verb-particle constructions. *SKY Journal of Linguistics*, 18(2005):145–173, 2005.

[9] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[10] Santanu Pal, Sudip Kumar Naskar, Pavel Pecina, Sivaji Bandyopadhyay, and Andy Way. Handling named entities and compound verbs in phrase-based statistical machine translation. Association for Computational Linguistics, 2010.

[11] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. *Proceedings of the Empiricial Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.

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

[13] Edson Marchetti da Silva and Renato Rocha Souza. Information retrieval system using multiwords expressions (mwe) as descriptors. *JISTEM-Journal of Information Systems and Technology Management*, 9(2):213–234, 2012.

[14] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics, 2012.

[15] Aline Villavicencio, Francis Bond, Anna Korhonen, and Diana McCarthy. Introduction to the special issue on multi-word expressions: Having a crack at a hard nut. *Computer Speech & Language*, 19(4):365–377, 2005.

[16] Veronika Vincze, István Nagy, and Gábor Berend. Multiword expressions and named entities in the wiki50 corpus. In *RANLP*, pages 289–295, 2011.