# Learning Paraphrasing for Multi-word Expressions

**Seid Muhie Yimam**[†] and **Héctor Martínez Alonso**[◇]
and **Martin Riedl**[†] and **Chris Biemann**[†]

[†]**FG Language Technology**
Computer Science Department
Technische Universität Darmstadt

[◇]**University of Paris 7**
The National Institute for Research
in Computer Science and Control – INRIA

## Abstract

In this paper, we investigate the impact of context for the paraphrase ranking task, comparing and quantifying results for multi-word expressions and single words. We focus on systematic integration of existing paraphrase resources to produce paraphrase candidates and later ask human annotators to judge paraphrasability in context.

We first conduct a paraphrase-scoring annotation task with and without context for targets that are i) single- and multi-word expressions ii) verbs and nouns. We quantify how differently annotators score paraphrases when context information is provided. Furthermore, we report on experiments with automatic paraphrase ranking. If we regard the problem as a binary classification task, we obtain an F1–score of 81.56% and 79.87% for multi-word expressions and single words resp. using kNN classifier. Approaching the problem as a learning-to-rank task, we attain MAP scores up to 87.14% and 91.58% for multi-word expressions and single words resp. using LambdaMART, thus yielding high-quality contextualized paraphrased selection. Further, we provide the first dataset with paraphrase judgments for multi-word targets in context.

## 1 Introduction

In this work, we examine the influence of context for paraphrasing of multi-word expressions (MWEs). Paraphrases are alternative ways of writing texts while conveying the same information (Zhao et al., 2007; Burrows et al., 2013). There are several applications where an automatic text paraphrasing is desired such as text shortening (Burrows et al., 2013), text simplification, machine translation (Kauchak and Barzilay, 2006), or textual entailment.

Over the last decade, a large number of paraphrase resources have been released including PPDB (Pavlick et al., 2015), which is the largest in size. However, PPDB provides only paraphrases without context. This hampers the usage of such a resource in applications. In this paper, we tackle the research question on how we can automatically rank paraphrase candidates from abundantly available paraphrase resources. Most existing work on paraphrases focuses on lexical-, phrase-, sentence- and document level (Burrows et al., 2013). We primarily focus on contextualization of paraphrases based on existing paraphrase resources.

Furthermore, we target multi-worded paraphrases, since single-word replacements are covered well in lexical substitution datasets, such as (McCarthy and Navigli, 2007; Biemann, 2012). While these datasets contain multi-word substitution candidates, the substitution targets are strictly single words. Multi-word expressions are prevalent in text, constituting roughly as many entries as single words in a speaker's lexicon (Sag et al., 2002), and are important for a number of NLP applications. For example, the work by Finlayson and Kulkarni (2011) shows that detection of multi-word expressions improves the F-score of a word sense disambiguation task by 5 percent. In this paper, we experiment with both MWE and single words and investigate the difficulty of the paraphrasing task for single words vs. MWEs, using the same contextual features.

Our work, centered in assessing the effect of context for paraphrase ranking of humans and its automatic prediction, includes the following steps: 1) systematic combination of existing paraphrase

*Proceedings of the 12th Workshop on Multiword Expressions*, pages 1–10,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

resources to produce paraphrase candidates for single- and multi-word expressions, 2) collection of dataset for paraphrase ranking/selection annotation task using crowdsourcing, and 3) investigating different machine learning approaches for an automatic paraphrase ranking.

## 2 Related Work

### 2.1 Paraphrase Resources and Machine Learning Approaches

Paraphrasing consists of mainly two tasks, paraphrase *generation* and paraphrase *identification*. Paraphrase generation is the task of obtaining candidate paraphrases for a given target. Paraphrase identification estimates whether a given paraphrase candidate can replace a paraphrase target without changing the meaning in context.

PPDB (Pavlick et al., 2015) is one of the largest collections of paraphrase resources collected from bilingual parallel corpora. PPDB2 has recently been released with revised ranking scores. It is based on human judgments for 26,455 paraphrase pairs sampled from PPDB1. They apply ridge regression to rank paraphrases, using the features from PPDB1 and include word embeddings.

The work of (Kozareva and Montoyo, 2006) uses a dataset of paraphrases that were generated using monolingual machine translation. In the dataset, sentence pairs are annotated as being paraphrases or not. For the binary classification, they use three machine learning algorithms (SVM, kNN and MaxEnt). As features they use word overlap features, n-grams ratios between targets and candidates, skip-grams longest common subsequences, POS tags and proper names.

Connor and Roth (2007) develop a global classifier that takes a word $v$ and its context, along with a candidate word $u$, and determines whether $u$ can replace $v$ in the given context while maintaining the original meaning. Their work focuses on verb paraphrasing. Notions of context include: being either subject or object of the verb, named entities that appear as subject or object, all dependency links connected to the target, all noun phrases in sentences containing the target, or all of the above.

The work of Brockett and Dolan (2005) uses annotated datasets and Support Vector Machines (SVMs) to induce larger monolingual paraphrase corpora from a comparable corpus of news clusters found on the World Wide Web. Features include morphological variants, WordNet synonyms and hypernyms, log-likelihood-based based word pairings dynamically obtained from baseline sentence alignments, and string features such as word-based edit distance

Bouamor et al. (2011) introduce a targeted paraphrasing system, addressing the task of rewriting of subpart of a sentence to make the sentences easier for automatic translation. They report on experiments of rewriting sentences from Wikipedia edit history by contributors using existing paraphrase resources and web queries. An SVM classifier has been used for evaluation and an accuracy of 70% has been achieved.

Using a dependency-based context-sensitive vector-space approach, Thater et al. (2009) compute vector-space representations of predicate meaning in context for the task of paraphrase ranking. An evaluation on the subset of SemEval 2007 lexical substitution task produces a better result than the state-of-the-art systems at the time.

Zhao et al. (2007) address the problem of context-specific lexical paraphrasing using different approaches. First, similar sentences are extracted from the web and candidates are generated based on syntactic similarities. Candidate paraphrases are further filter using POS tagging. Second, candidate paraphrases are validated using different similarity measures such as co-occurrence similarity and syntactic similarity.

Our work is similar to previous approaches on all-words lexical substitution (Szarvas et al., 2013; Kremer et al., 2014; Hintz and Biemann, 2016) in the sense that we construct delexicalized classifiers for ranking paraphrases: targets, paraphrase candidates and context are represented without lexical information, which allows us to learn a single classifier/ranker for all potential paraphrasing candidates. However, these approaches are limited to single-word targets (Szarvas et al., 2013) resp. single-word substitutions (Kremer et al., 2014) only. In this paper, we extend these notions to MWE targets and substitutions, highlight the differences to single-word approaches, and report both on classification and ranking experiments.

### 2.2 Multi-word Expression Resources

While there are some works on the extraction of multi-word expressions and on investigation of their impact on different NLP applications, as far as we know, there is no single work dedicated on paraphrasing multi-word expressions. Various approaches exist for the extraction of MWEs: Tsvetkov and Wintner (2010) present an approach to extract MWEs from parallel corpora. They align the parallel corpus and focus on misalignment, which typically indicates expressions in the source language that are translated to the target in a non-compositional way. Frantzi et al. (2000) present a method to extract multi-word terms from English corpora, which combines linguistic and statistical information. The Multi-word Expression Toolkit (*MWEtoolkit*) extracts MWE candidates based on flat n-grams or specific morphosyntactic patterns (of surface forms, lemmas, POS tags) (Ramisch et al., 2010) and apply different fillters ranging form simple count thresholds to a more complex cases such as Association Measures (AMs). The tool further supports indexing and searching of MWEs, validation, and annotation facilities.

Schneider et al. (2014) developed a sequence-tagging-based supervised approach to MWE identification. A rich set of features has been used in a linguistically-driven evaluation of the identification of heterogeneous MWEs. The work by Vincze et al. (2011) constructs a multi-word expression corpus annotated with different types of MWEs such as compound, idiom, verb-particle constructions, light verb constructions, and others. In our work, we have used a combination of many MWEs resources from different sources for both MWE target detection and candidate generation (see Subsection 3.2).

## 3 Methods

In this section we describe our approach, which covers: the collection of training data, detection of multi-word paraphrases including annotating substitutes and learning a classifier in order to rank substitute candidates for a target paraphrase.

### 3.1 Impact of Context on Paraphrasing

In order to validate our intuitively plausible hypothesis that context has an impact on paraphrasing, we conduct experiments using the PPDB2 paraphrase database. PPDB2 is released with better paraphrase ranking than PPDB1 (Pavlick et al., 2015) but does not incorporate context information. Hence, we carry out different paraphrase ranking and selection annotation tasks using the Amazon Mechanical Turk crowdsourcing platform.

| | All ($\rho$) | MWE ($\rho$) | Single ($\rho$) |
|---|---|---|---|
| No context | 0.35 | 0.25 | 0.36 |
| Context | 0.31 | 0.23 | 0.32 |

Table 1: Spearman correlation of human judgment with PPDB2 default rankings. The column *MWE* shows the result of only MWEs and the column *Single* shows the result of only single words.

In the first annotation task, a total of 171 sentences are selected from the British Academic Written English (BAWE) corpus[1] (Alsop and Nesi, 2009), with five paraphrase targets. The targets are selected in such a way that a) include MWEs as targets when it is possible (see Subection 3.2 how we select targets), b) the candidates could bear more than one contextual meaning and, c) workers can select up to three paraphrases and have to supply their own paraphrase if none of the candidates match. To satisfy condition b), we have used the JoBimText DT database API (Ruppert et al., 2015) to obtain single word candidates with multiple senses according to automatic sense induction.

We conduct this annotation setup twice, both with and without showing the original context (3–8 sentences). For both setups, a task is assigned to 5 workers. We incorporate control questions with invalid candidate paraphrases in order to reject unreliable workers. In addition to the control questions, JavaScript functions are embedded to ensure that workers select or supply at least one paraphrase. The results are aggregated by summing the number of workers that agreed on candidates, for scores between 0 and 5. Table 1 shows the Spearman correlation results. We can see that both single and MWE targets are context-dependent, as correlations are consistently lower when taking context into account. Further, we note that correlations are positive, but low, indicating that the PPDB2 ranking should not be used as-is for paraphrasing.

### 3.2 Paraphrase Dataset Collection using Crowdsourcing

In this subsection, we present the processes carried out to collect datasets for the paraphrase ranking task. This includes selection of documents,

[1] https://www2.warwick.ac.uk/fac/soc/al/research/collections/bawe/

identification of target paraphrases, and generation of candidate paraphrases from existing resources. We use 2.8k essay sentences from the ANC[2] and BAWE corpora for the annotation task.

**Target detection and candidate generation:** In order to explore the impact of contexts for paraphrasing, the first step is to determine possible targets for paraphrasing, as shown in Figure 1. As a matter of fact, every word or MWE in a sentence can be a target for paraphrasing. When prototyping the annotation setup, we found that five paraphrase targets are a reasonable amount to be completed in a single Human Intelligence Task (HIT), a single and self-contained unit of task to be completed and submitted by an annotator to receive a reward in a return[3].

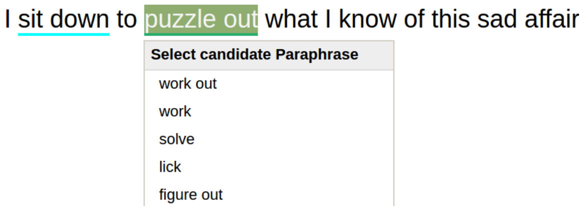

I sit down to puzzle out what I know of this sad affair

Figure 1: Paraphrase targets (a) and paraphrase candidates (b).

We select targets that have at least five candidates in our combined paraphrase resources. The paraphrase resources (*S*) for candidates generations are composed of collections from PPDB (Pavlick et al., 2015), WordNet and JoBimText distributional thesaurus (DT – only for single words).

For MWE paraphrase targets, we have used different MWE resources. A total of 79,349 MWE are collected from WordNet, STREUSLE (Schneider and Smith, 2015; Schneider et al., 2014)[4], Wiki50 (Vincze et al., 2011) and the MWE project (McCarthy et al., 2003; Baldwin and Villavicencio, 2002)[5]. We consider MWEs from this resources to be a paraphrase target when it is possible to generate paraphrase candidates from our paraphrase resources (*S*).

Candidates paraphrases for a target (both single and MWE) are generated as follows. For each paraphrase target, we retrieve candidates from the resources (*S*). When more than five candidates are collected: 1) for single words, we select the top candidates that bear different meanings in context using the automatic sense induction API by Ruppert et al. (2015), 2) for MWEs we select candidates that are collected from multiple resources in *S*. We present five candidates for the workers to select the suitable candidates in context. We also allow workers to provide their own alternative candidates when they found that none of the provided candidates are suitable in the current context. Figure 2 shows the Amazon Mechanical Turk user interface for the paraphrase candidate selection task. We discuss the different statistics and quality of annotations obtained in Section 5.2.

### 3.3 Machine Learning Approaches for Paraphrasing

In this work we investigate two types of machine-learning setups for paraphrase selection and ranking problems. In the first setup, we tackle the problem as a binary classification task, namely whether one candidate can be chosen to replace a target in context. All candidates annotated as possible paraphrases are considered a positive examples. We follow a 5-fold cross validation approach to train and evaluate our model.

In the second setup, we use a learning-to-rank algorithm to re-rank paraphrase candidates. There are different machine learning methods for the learning-to-ranking approach, such as *pointwise*, *pairwise* and *listwise* rankings. In the pointwise ranking, a model is trained to map candidate phrases to relevance scores, for example using a simple regression technique. Ranking is then performed by simply sorting predicted scores (Li et al., 2007). In the pairwise approach, the problem is regarded as a binary classification task where pairs are individually compared each other (Freund et al., 2003). Listwise ranking approaches learn a function by taking individual candidates as instances and optimizing a loss function defined on the predicted instances (Xia et al., 2008). We experiment with different learning-to-rank algorithms from the RankLib[6] Java package of the Lemur project[7]. In this paper, we present the results obtained using LambdaMART. LambdaMART (Burges, 2010) uses gradient boosting

[2] http://www.anc.org/

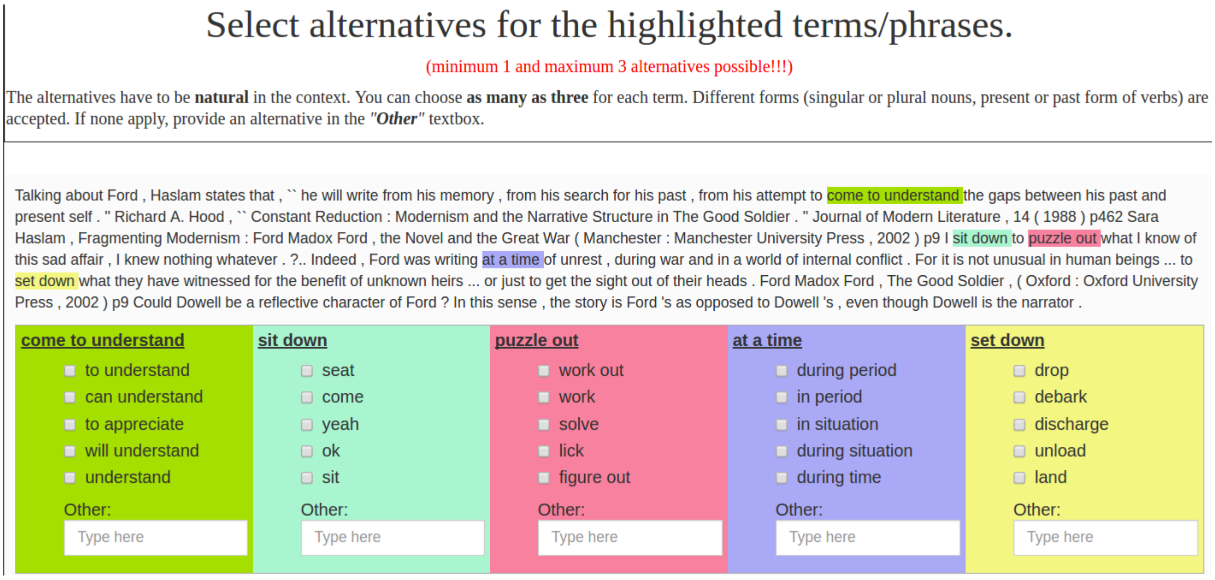
[3] https://www.mturk.com/mturk/help?helpPage=overview

[4] http://www.cs.cmu.edu/~ark/LexSem/

[5] http://mwe.stanford.edu

[6] https://people.cs.umass.edu/~vdang/ranklib.html

[7] http://sourceforge.net/projects/lemur/



Select alternatives for the highlighted terms/phrases.

(minimum 1 and maximum 3 alternatives possible!!!)

The alternatives have to be **natural** in the context. You can choose **as many as three** for each term. Different forms (singular or plural nouns, present or past form of verbs) are accepted. If none apply, provide an alternative in the ***"Other"*** textbox.

Talking about Ford , Haslam states that , `` he will write from his memory , from his search for his past , from his attempt to come to understand the gaps between his past and present self . " Richard A. Hood , `` Constant Reduction : Modernism and the Narrative Structure in The Good Soldier . " Journal of Modern Literature , 14 ( 1988 ) p462 Sara Haslam , Fragmenting Modernism : Ford Madox Ford , the Novel and the Great War ( Manchester : Manchester University Press , 2002 ) p9 I sit down to puzzle out what I know of this sad affair , I knew nothing whatever . ?.. Indeed , Ford was writing at a time of unrest , during war and in a world of internal conflict . For it is not unusual in human beings ... to set down what they have witnessed for the benefit of unknown heirs ... or just to get the sight out of their heads . Ford Madox Ford , The Good Soldier , ( Oxford : Oxford University Press , 2002 ) p9 Could Dowell be a reflective character of Ford ? In this sense , the story is Ford 's as opposed to Dowell 's , even though Dowell is the narrator .

| come to understand | sit down | puzzle out | at a time | set down |
|---|---|---|---|---|
| ☐ to understand | ☐ seat | ☐ work out | ☐ during period | ☐ drop |
| ☐ can understand | ☐ come | ☐ work | ☐ in period | ☐ debark |
| ☐ to appreciate | ☐ yeah | ☐ solve | ☐ in situation | ☐ discharge |
| ☐ will understand | ☐ ok | ☐ lick | ☐ during situation | ☐ unload |
| ☐ understand | ☐ sit | ☐ figure out | ☐ during time | ☐ land |
| Other: Type here | Other: Type here | Other: Type here | Other: Type here | Other: Type here |

Figure 2: User-interface for paraphrase selection.

to directly optimize learning-to-rank specific cost functions such as Normalized Discounted Cumulative Gain (NDCG) and Mean Average Precision (MAP).

### 3.4 Features

We have modeled three types of features: a *resource-based* feature where feature values are taken from a lexical resource ($F0$), four features based on *global context* where we use word embeddings to characterize targets and candidates irrespectively of context ($F1, 2, 3, 4$) and four features based on *local context* that take the relation of target and candidate with the context into account ($F5, 6, 7, 8$).

**PPDB2 score**: We use the the PPDB2 score ($F0$) of each candidate as baseline feature. This score reflects a context-insensitive ranking as provided by the lexical resources.

First we describe features considering global context information:

**Target and Candidate phrases**: Note that we do not use word identity as a feature, and use the word embedding instead for the sake of robustness. We use the word2vec python implementation of Gensim (Řehůřek and Sojka, 2010)[8] to generate embeddings from BNC[9], Wikipedia, BAWE and ANC. We train embeddings with 200 dimensions using skip-gram training and a window size of 5. We approximate MWE embeddings by averaging the embeddings of their parts. We use the word embeddings of the target ($F1$) and the candidate ($F2$) phrases.

**Candidate-Target similarities**: The dot product of the target and candidate embeddings ($F3$), as described in (Melamud et al., 2015).

**Target-Sentence similarity**: The dot product between a candidate and the sentence, i.e. the average embeddings of all words in the sentence ($F4$).

The following features use local context information:

**Target-Close context similarity**: The dot product between the candidate and the left and right 3-gram ($F5$) and 5-gram embedding ($F6$) resp..

**Ngram features**: A normalized frequency for a 2-5-gram context with the target and candidate phrases ($F7$) based on Google Web 1T 5-Grams[10].

**Language model score**: A normalized language model score using a sentence as context with the target and candidate phrases ($F8$). An n-gram language model (Pauls and Klein, 2011) is built using the BNC and Wikipedia corpora.

Also, we experimented with features that eventually did not improve results, such as the embeddings of the target's $n = 5$ most similar words, length and length ratios between target and candidate, most similar words and number of shared senses among target and candidate phrases based JoBimText DT (Ruppert et al., 2015), and N-gram POS sequences and dependency labels of the tar-

[8] https://radimrehurek.com/gensim/models/word2vec.html

[9] http://www.natcorp.ox.ac.uk/

[10] https://catalog.ldc.upenn.edu/LDC2009T25

get.

| | kNN | | | LambdaMART | | |
|---|---|---|---|---|---|---|
| Features | P | R | F | P@1 | NDCG @5 | MAP |
| All | 69.27 | 90.41 | 78.41 | 90.53 | 89.03 | **91.35** |
| F0+1+2+5 | 76.14 | 84.40 | 80.04 | 89.38 | 89.24 | 91.31 |
| F1+2 | 75.28 | 85.05 | 79.85 | 88.13 | 88.98 | 90.88 |
| F1+3 | 75.28 | 85.05 | 79.85 | 88.13 | 88.98 | 90.88 |
| F1+5 | 74.42 | 86.69 | **80.07** | 88.11 | 88.76 | 90.82 |
| F0+1+2+7 | 74.89 | 85.65 | 79.89 | 89.42 | 89.34 | 91.29 |
| F3+7 | 70.28 | 79.82 | 74.61 | 82.31 | 84.08 | 86.34 |
| F5+7 | 64.56 | 86.25 | 73.64 | 80.24 | 82.61 | 85.60 |
| F0+3 | 68.87 | 81.39 | 74.43 | 87.04 | 86.37 | 88.78 |
| F0+7 | 69.86 | 79.02 | 74.05 | 84.14 | 84.69 | 87.20 |
| F6+7 | 65.20 | 79.49 | 71.34 | 80.03 | 84.98 | 85.54 |
| F0+6 | 67.43 | 78.04 | 72.08 | 84.98 | 85.26 | 87.64 |
| F0 | 72.49 | 79.84 | 75.18 | 84.12 | 84.51 | 87.15 |

(a) Performance on all datasets

| | kNN | | | LambdaMART | | |
|---|---|---|---|---|---|---|
| Features | P | R | F | P@1 | NDCG @5 | MAP |
| All | 76.74 | 82.99 | 79.71 | 89.72 | 88.82 | **91.58** |
| F0+1+2+5 | 75.36 | 84.54 | 79.67 | 90.38 | 89.10 | 91.41 |
| F1+2 | 75.74 | 83.66 | 79.49 | 88.28 | 88.82 | 90.98 |
| F1+3 | 75.74 | 83.66 | 79.49 | 88.28 | 88.82 | 90.98 |
| F1+5 | 74.95 | 85.52 | **79.87** | 87.50 | 88.51 | 90.76 |
| F0+1+2+7 | 69.59 | 88.63 | 77.95 | 90.00 | 89.31 | 91.49 |
| F3+7 | 70.25 | 78.71 | 74.09 | 81.92 | 83.78 | 86.03 |
| F5+7 | 64.05 | 85.20 | 72.90 | 79.96 | 82.24 | 85.09 |
| F0+3 | 68.89 | 80.52 | 74.05 | 86.41 | 86.46 | 88.64 |
| F0+7 | 69.93 | 78.38 | 73.77 | 84.14 | 84.77 | 87.11 |
| F6+7 | 64.67 | 78.80 | 70.71 | 78.97 | 82.06 | 84.98 |
| F0+6 | 66.98 | 77.28 | 71.44 | 85.21 | 85.04 | 87.55 |
| F0 | 74.08 | 72.18 | 71.47 | 84.81 | 84.60 | 87.29 |

(b) Performance on single words datasets

| | kNN | | | LambdaMART | | |
|---|---|---|---|---|---|---|
| Features | P | R | F | P@1 | NDCG @5 | MAP |
| All | 69.81 | 95.70 | 80.60 | 84.69 | 77.54 | 86.21 |
| F0+1+2+5 | 73.66 | 91.25 | **81.56** | 81.76 | 76.40 | 85.43 |
| F1+2 | 73.25 | 91.11 | 81.13 | 82.74 | 76.00 | 86.69 |
| F1+3 | 73.25 | 91.11 | 81.13 | 82.74 | 76.00 | 86.69 |
| F1+5 | 72.58 | 92.05 | 81.05 | 84.69 | 77.14 | **87.14** |
| F0+1+2+7 | 72.85 | 91.14 | 80.89 | 83.71 | 75.95 | 84.97 |
| F3+7 | 71.56 | 85.18 | 77.57 | 78.83 | 72.71 | 80.40 |
| F5+7 | 68.03 | 89.72 | 77.18 | 72.31 | 67.27 | 80.66 |
| F0+3 | 70.05 | 85.64 | 76.91 | 81.43 | 71.32 | 81.62 |
| F0+7 | 70.28 | 84.56 | 76.56 | 71.34 | 67.76 | 77.35 |
| F6+7 | 69.46 | 85.38 | 76.45 | 79.48 | 67.82 | 79.66 |
| F0+6 | 71.49 | 82.35 | 76.39 | 80.78 | 69.16 | 82.37 |
| F0 | 73.35 | 70.54 | 69.06 | 69.71 | 67.12 | 77.95 |

(c) Performance on MWEs datasets

Table 2: Binary classification vs. learning-to-rank results on baseline and 8 top-performing feature combinations.

## 4 Experimental Results

Now we discuss the different experimental results using the K-Nearest Neighbors (kNN)[11] from the scikit-learn[12] machine leaning framework (binary classification setup) and the LambdaMART learning to rank algorithm from the RankLib (learning to rank setup). We have used 5-fold cross validation on 17k data points (2k MWEs and 15k single) from the crowdsourcing annotation task for both approaches. The cross-validation is conducted in a way that there is no target overlap in in each split, so that our model is forced to learn a delexicalized function that can apply to all targets where substitution candidates are available, cf. (Szarvas et al., 2013).

As evaluation metrics, precision, recall, and F-score are used for the first setup. For the second setup we use P@1, Mean Average Precision (MAP), and Normalized Discounted Cumulative Gain (NDCG). P@1 measures the percentage of correct paraphrases at rank 1, thus gives the percentage of how often the best-ranked paraphrase is judged as correct. MAP provides a single-figure measure of quality across recall levels. NDCG is a ranking score that compares the optimal ranking to the system ranking, taking into account situations where many resp. very few candidates are relevant (Wang et al., 2013). In the following subsections, we will discuss the performance of the two machine learning setups.

### 4.1 Binary Classification

For paraphrase selection, we regard the problem as a binary classification task. If a given candidate is selected by at least one annotator, it is considered as possible substitute and taken as positive example. Otherwise it will be considered as a negative training example. For this experiment, kNN from the scikit-learn machine learning framework is used. Table 2 shows the evaluation result for the best subsets of feature combinations. The classification experiments obtain maximal F1s of 81.56% for MWEs and 79.77% for single words vs. a non-contextual baseline of 69.06% and 71.47% resp.

[11] Parameters: Number of neighbors (n_neighbors) = 20, weight function (weights) = distance

[12] http://scikit-learn.org/

### 4.2 Learning to Rank

Now we learn to rank paraphrase candidates, using the number of annotators agreeing on each candidate to assign relevance scores in the interval of [0–5].. The average evaluation result on the 5-fold splits is shown in Table 2. The baseline ranking given by $F0$ is consistently lower than our context-aware classifiers. The best scores are attained with all features enabled (P@1=89.72, NDCG@5=88.82 and MAP=91.58 for single words vs. P@1=84.69, NDCG@5=77.54 and MAP=86.21 for MWEs). A more detailed analysis between the ranking of single-worded targets and multi-worded paraphrases will be discussed in Section 5.3.

## 5 Analysis of the Result

In this section, we interpret the results obtained during the crowdsourcing annotation task and machine learning experimentation.

### 5.1 Correlation with PPDB2 Ranking

As it can be seen from Table 1, without contexts, a Spearman correlation of 0.36 and 0.25 is obtained by the workers against the PPDB2 default rankings for single and MWE annotations resp. However, when the contexts are provided to the workers, the ranking for the same items is lower with a Spearman correlation of 0.32 and 0.23 for single and MWE annotations resp. This indicates that the contexts provided has an impact on the ranking of paraphrases. Moreover, we observe that the correlation with PPDB2 ranking is considerably lower than the one reported by Pavlick et al. (2015) which is 0.71. Data analysis revealed a lot of inconsistent scores within the PPDB2. For example, the word pairs (*come in*, *sound*) and (*look at*, *okay*) have a high correlation score (3.2, 3.18 resp.). However, they do not seem to be related and are not considered as substitutable by our method. The perceived inconsistency is worse in the case of MWE scores hence the correlation is lower than for single words.

### 5.2 Annotation Agreement

|  | #0 | #1 | #2 | #3 | #4 | #5 | Agreement |
|---|---|---|---|---|---|---|---|
| All | 36.09 | 34.57 | 11.68 | 8.38 | 5.82 | 3.46 | 81.56 |
| Single | 36.54 | 34.47 | 11.48 | 8.24 | 5.79 | 3.48 | 81.76 |
| MWE | 32.39 | 35.43 | 13.35 | 9.47 | 6.06 | 3.30 | 76.97 |

Table 3: Score distributions and observed annotation agreement (in %). The columns #1 to #5 shows the percentage of scores the annotator give to each classes (0–5). The last column provides the observed agreements among 5 annotators.

According to Table 3, annotators agree more often on single words than on MWEs. This might be attributed to the fact that single word candidates are generated with different meanings using the automatic sense induction approach, provided by the JoBimText framework (Ruppert et al., 2015). Hence, when context is provided, it is much easier to discern the correct candidate paraphrase. On the other hand, in MWEs, their parts disambiguate each other to some extent, so there are less candidates with context mismatches. We can witness that from the individual class percentages (MWE candidates are on average scored higher than single word candidates, especially in the range of [2-4]) and from the overall observed agreements.

### 5.3 Machine Learning

According to the results shown in Table 2, we achieve higher scores for the binary classification for MWE than for single words. We found that this is due to the fact that we have more positive examples (67.6%) than the single words. Intuitively, it is much easier to have one of the five candidates to be a correct paraphrase as most of the MWE are not ambiguous in meaning (see recall (R) column in Table 2).

**Example 1**: *this is the reason too that the reader disregards the duke 's* ***point of view*** *, and supports and sympathises with the duchess , acknowledging her innocence.*
**Example 2**: *this list of verbs describes day-to-day occupations of the* ***young girl*** *, suggesting that she does n't distinguish the graveyard from other locations of her day .*
**Example 3**: *this is apparent in the case of the priest who tries to vanquish* ***the devil*** *, who is infact mistaken for mouse slayer , the cat ...*

Error analysis of the classification result shows that some of the errors are due to annotation mistakes. In Example 1, the annotators do not select the candidate **stand** while the classifier predicts it correctly. We also found that the classifier wrongly picks antonyms from candidates. The classifier selected **younger man** and **heaven** for Example 2 and 3 resp. while the annotators do not

| Target | Candidate | #Annotators | Ranker score |
|---|---|---|---|
| write about | write on | 2 | 8.14 |
| write about | write into | 0 | 5.63 |
| write about | discuss | 1 | 2.81 |
| write about | write in | 1 | 1.20 |
| write about | talk to | 1 | -1.82 |

Table 4: LambdaMART ranking scores

select them. Out of 91 MWE examples predicted by the classifier as positive, we found out that 24 of the examples have near synonym meaning while annotators fail to select them and also, 7 examples are antonyms.

The results for learning the ranking show a different trend. Once again, we can see that it is difficult to rank better when the candidates provided (in the case of MWEs) are less ambiguous. This could also be a consequence of the lower agreement on MWE candidate judgments. Analysis of the learn-to-rank result also revealed that the lower result is due to the fact that more often, the annotators do not agree on a single candidate, as it can be seen from Table 4.

Looking at the overall results, it becomes clear that our learning framework can substantially improve contextual paraphrase ranking over the PPDB2-resource-based baseline. The resource-based $F0$-feature, however, is still important for attaining the highest scores. While the global context features based on word embeddings (cf. $F1+2+3$ or $F1+3$) already show a very good performance, they are consistently improved by adding one or all feature that models local context ($F5, F6, F7, F8$). From this we conclude that all feature types (resource, global context, local context) are important.

## 6 Conclusion and Future Directions

In this paper we have quantified the impact of context on the paraphrase ranking scoring task. The direct annotation experiments show that paraphrasing is in fact a context-specific task: while the paraphrase ranking scores provided by PPDB2 were confirmed by a weak correlation with out-of-context judgments, the correlation between resource-provided rankings and judgments in context were consistently lower.

We conducted a classification experiment in a delexicalized setting, i.e. training and testing on disjoint sets of paraphrase targets. For a binary classification setting as well as for ranking, we improved substantially over the non-contextualized baseline as provided by PPDB2. An F-score of 81.56% and 79.87% is attained for MWEs and Single words using kNN classifier from scikit-learn. A MAP score of 87.14% and 91.58% is obtained for MWEs and single words using the LambdaMART learn-to-rank algorithm from RankLib.

We recommend to use a learning-to-rank framework for utilizing features that characterize the paraphrase candidate not only with respect to the target, but also with respect to the context. The most successful features in these experiments are constructed from word embeddings, and the best performance is attained in combination of resource-based, global context and local context features.

Both experiments confirm the generally accepted intuition that paraphrasing, just like lexical substitution of single words, depends on context: while MWEs are less ambiguous than single words, it still does not hold that they can be replaced without taking the context into account. Here, we have quantified the amount of context dependence on a new set of contextualized paraphrase judgments, which is – to our knowledge – the first dataset with multi-word targets[13].

While our dataset seems of sufficient size to learn a high-quality context-aware paraphrase ranker, we would like to employ usage data from a semantic writing aid for further improving the quality, as well as for collecting domain- and user-specific paraphrase generation candidates.

## References

Sian Alsop and Hilary Nesi. 2009. Issues in the development of the British Academic Written English (BAWE) corpus. *Corpora*, 4(1):71–83.

Timothy Baldwin and Aline Villavicencio. 2002. Extracting the unextractable: A case study on verb-particles. In *Proceedings of the 6th Conference on Natural Language Learning*, CoNLL-02, pages 1–7, Taipei, Taiwan.

Chris Biemann. 2012. Creating a System for Lexical Substitutions from Scratch using Crowdsourcing. *Language Resources and Evaluation: Special Issue on Collaboratively Constructed Language Resources*, 46(2):97–112.

[13] The AMT judgment datasets are provided as supplementary material and will be distributed under CC-BY.

Houda Bouamor, Aurélien Max, Gabriel Illouz, and Anne Vilnat. 2011. Web-based validation for contextual targeted paraphrasing. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 10–19, Portland, OR, USA.

Chris Brockett and William B. Dolan. 2005. Support vector machines for paraphrase identification and corpus construction. In *Third International Workshop on Paraphrasing (IWP2005)*, pages 1–8, Jeju Island, South Korea.

Christopher J.C. Burges. 2010. From ranknet to lambdarank to lambdamart: An overview. Technical Report MSR-TR-2010-82, Microsoft Research.

Steven Burrows, Martin Potthast, and Benno Stein. 2013. Paraphrase acquisition via crowdsourcing and machine learning. *ACM Trans. Intell. Syst. Technol.*, pages 43:1–43:21.

Michael Connor and Dan Roth. 2007. Context sensitive paraphrasing with a single unsupervised classifier. In *18th European Conference on MAchine Learning (ECML)*, pages 289–295, Warsaw, Poland.

Mark Alan Finlayson and Nidhi Kulkarni. 2011. Detecting multi-word expressions improves word sense disambiguation. In *Proceedings of the Workshop on Multiword Expressions: From Parsing and Generation to the Real World*, MWE '11, pages 20–24, Portland, OR, USA.

Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the C-value/NC-value method. *International Journal on Digital Libraries*, 3(2):115–130.

Yoav Freund, Raj Iyer, Robert E Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *The Journal of machine learning research*, 4:933–969.

Gerold Hintz and Chris Biemann. 2016. Language Transfer Learning for Supervised Lexical Substitution. In *The Annual Meeting of the Association for Computational Linguistics (ACL)*, page to apear, Berlin, Germany.

David Kauchak and Regina Barzilay. 2006. Paraphrasing for automatic evaluation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, HLT-NAACL '06, pages 455–462, New York, NY, USA.

Zornitsa Kozareva and Andrés Montoyo. 2006. Paraphrase identification on the basis of supervised machine learning techniques. In *Advances in natural language processing*, pages 524–533, Turku, Finland.

Gerhard Kremer, Katrin Erk, Sebastian Padó, and Stefan Thater. 2014. What substitutes tell us - analysis of an "all-words" lexical substitution corpus. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 540–549, Gothenburg, Sweden.

Ping Li, Qiang Wu, and Christopher J Burges. 2007. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in neural information processing systems*, pages 897–904, Vancouver, BC, Canada.

Diana McCarthy and Roberto Navigli. 2007. Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 48–53, Prague, Czech Republic.

Diana McCarthy, Bill Keller, and John Carroll. 2003. Detecting a continuum of compositionality in phrasal verbs. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 73–80, Sapporo, Japan.

Oren Melamud, Omer Levy, and Ido Dagan. 2015. A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 1–7, Denver, CO, USA.

Adam Pauls and Dan Klein. 2011. Faster and smaller n-gram language models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 258–267, Portland, OR, USA.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2015. PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 425–430, Beijing, China.

Carlos Ramisch, Aline Villavicencio, and Christian Boitet. 2010. MWEtoolkit: a Framework for Multiword Expression Identification. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, pages 134–136, Valletta, Malta.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta.

Eugen Ruppert, Manuel Kaufmann, Martin Riedl, and Chris Biemann. 2015. JOBIMVIZ: A Web-based Visualization for Graph-based Distributional Semantic Models. In *The Annual Meeting of the Association for Computational Linguistics (ACL) System Demonstrations*, pages 103–108, Beijing, China.

Ivan A. Sag, Timothy Baldwin, Francis Bond, Ann A. Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing '02, pages 1–15, London, UK.

Nathan Schneider and Noah A. Smith. 2015. A corpus and model integrating multiword expressions and supersenses. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1537–1547, Denver, CO, USA.

Nathan Schneider, Emily Danchik, Chris Dyer, and Noah Smith. 2014. Discriminative Lexical Semantic Segmentation with Gaps: Running the MWE Gamut. *Transactions of the Association for Computational Linguistics*, 2:193–206.

György Szarvas, Róbert Busa-Fekete, and Eyke Hüllermeier. 2013. Learning to rank lexical substitutions. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1926–1932, Seattle, WA, USA.

Stefan Thater, Georgiana Dinu, and Manfred Pinkal. 2009. Ranking paraphrases in context. In *Proceedings of the 2009 Workshop on Applied Textual Inference*, TextInfer '09, pages 44–47, Suntec, Singapore.

Yulia Tsvetkov and Shuly Wintner. 2010. Extraction of multi-word expressions from small parallel corpora. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 1256–1264, Beijing, China.

Veronika Vincze, István Nagy T., and Gábor Berend. 2011. Multiword Expressions and Named Entities in the Wiki50 Corpus. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 289–295, Hissar, Bulgaria.

Yining Wang, Liwei Wang, Yuanzhi Li, Di He, Wei Chen, and Tie-Yan Liu. 2013. A theoretical analysis of NDCG ranking measures. In *Proceedings of the 26th Annual Conference on Learning Theory (COLT 2013)*, Princeton, NJ, USA.

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. 2008. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, New York, NY, USA.

Shiqi Zhao, Ting Liu, Xincheng Yuan, Sheng Li, and Yu Zhang. 2007. Automatic acquisition of context-specific lexical paraphrases. In *International Joint Conference on Artificial Intelligence*, Hyderabad, India.