Leveraging Crowdsourcing for Paraphrase Recognition

Martin Tschirsich
Department of Computer Science,
TU Darmstadt
m.tschirsich@gmx.de

Gerold Hintz
Department of Computer Science,
TU Darmstadt
gerold.hintz@googlemail.com

Abstract

Crowdsourcing, while ideally reducing both costs and the need for domain experts, is no all-purpose tool. We review how paraphrase recognition has benefited from crowdsourcing in the past and identify two problems in paraphrase acquisition and semantic similarity evaluation that can be solved by employing a smart crowdsourcing strategy. First, we employ the CrowdFlower platform to conduct an experiment on sub-sentential paraphrase acquisition with early exclusion of low-accuracy crowdworkers. Second, we compare two human intelligence task designs for evaluating phrase pairs on a semantic similarity scale. While the first experiment confirms our strategy successful at tackling the problem of missing gold in paraphrase generation, the results of the second experiment suggest that, for both semantic similarity evaluation on a continuous and a binary scale, querying crowdworkers for a semantic similarity value on a multi-grade scale yields better results than directly asking for a binary classification.

1 Introduction

Paraphrase recognition[1] means to analyse whether two texts are paraphrastic, i.e. "a pair of units of text deemed to be interchangeable" (Dras, 1999). It has numerous applications in information retrieval, information extraction, machine translation and plagiarism detection. For instance, an internet search provider could recognize *"murder of the 35th U.S. president"* and *"assassination of John F. Kennedy"* to be paraphrases of each other and thus yield the same result. Paraphrase recognition is an open research problem and, even though having progressed immensely in recent years (Socher et al., 2011), state of the art performance is still below the human reference.

In this research, we analyse how *crowdsourcing* can contribute to paraphrase recognition. Crowdsourcing is the process of outsourcing a vast number of small, simple tasks, so called *HITs*[2], to a distributed group of unskilled workers, so called *crowdworkers*[3]. Reviewing current literature on the topic, we identify two problems in paraphrase acquisition and semantic similarity evaluation that can be solved by employing a smart crowdsourcing strategy. First, we propose how to reduce paraphrase generation costs by early exclusion of low-accuracy crowdworkers. Second, we compare two HIT designs for evaluating phrase pairs on a continuous semantic similarity scale. In order to evaluate our crowdsourcing strategies, we conduct our own experiments via the CrowdFlower[4] platform.

The rest of the paper is structured as follows. Section 2 first gives an overview of related work and lines out current approaches. We then proceed to our own experiments on crowdsourcing paraphrase acquisition (3.3) and semantic similarity evaluation (3.4). Section 4 and 5 conclude the study and propose future work in the area of paraphrase recognition and crowdsourcing.

2 Literature Review

Many research fields rely on paraphrase recognition and contribute to it, as there are many related concepts. These include inference rule discovery for question-answering and information retrieval (Lin and Pantel, 2001), idiom or multiword ex-

[1] the terms *paraphrase detection* and *paraphrase identification* might be used instead

[2] *Human Intelligence Tasks*

[3] often referred to as *turkers*

[4] http://crowdflower.com

Proceedings of the 7th Linguistic Annotation Workshop & Interoperability with Discourse, pages 205–213,
Sofia, Bulgaria, August 8-9, 2013. ©2013 Association for Computational Linguistics

pression acquisition (Fellbaum et al., 2006) and identification (Boukobza and Rappoport, 2009), machine translation evaluation (Snover et al., 2009), textual entailment recognition, and many more.

2.1 Paraphrase Definition

The notion of a paraphrase is closely related to the concepts of *semantic similarity* and *word ontology* and an exact definition is not trivial. Often, complex annotation guidelines and aggregated expert agreements decide whether phrases are to be considered paraphrastic or not (Dolan and Brockett, 2005). Formal definitions based e.g. on a domain theory and derivable facts (Burrows et al., 2013) have little practical relevance in paraphrase recognition. In terms of the semantic similarity relations *'equals'*, *'restates'*, *'generalizes'*, *'specifies'* and *'intersects'* (Marsi and Krahmer, 2010), *'paraphrase'* is equated with *'restates'*.

It is important to note that in the context of crowdsourcing, we, as well as most authors, rely on the crowdworker's intuition of what a paraphrase is. Usually, only a limited list of examples of desired valid paraphrases is given to the crowdworker as a reference.

2.2 Paraphrase Recognition

According to Socher et al. (2011), paraphrase recognition "determines whether two phrases of arbitrary length and form capture the same meaning". Paraphrase recognition is mostly understood as a binary classification process, although recently, some authors proposed a continuous semantic similarity measure (Madnani et al., 2012).

Competing paraphrase recognition approaches are often compared by their performance on the Microsoft Research Paraphrase Corpus (MSRPC). Until 2011, simple features such as n-gram overlap, dependency tree overlap as well as dependency tree edit distance produced the best results in terms of accuracy and F-measure values. However, algorithms based solely on such features can not identify semantic equivalence of synonymous words or phrases. Therefore, some authors subsequently integrated Wordnet synonyms as well as other corpus-based semantic similarity measures. The work of Madnani et al. (2012) based on the TERp machine translation evaluation metric (Snover et al., 2009) using synonyms and subsentential paraphrases presents the current state of the art for paraphrase detection on the MSRPC with an accuracy of 77.4% and F-measure of 84.1%. The inter-rater agreement amongst human annotators of 84% on the MSRPC can be considered as an upper bound for the accuracy that could be obtained using automatic methods (Fernando and Stevenson, 2008).

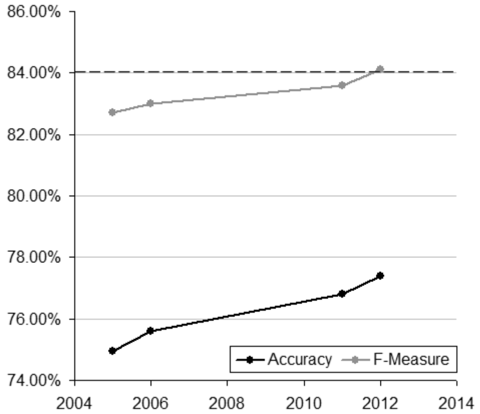


Figure 1: Highest ranking accuracy and F-measure over time for paraphrase recognition on the MSRPC with an inter-rater agreement amongst human annotators of 84%

As has become apparent, modern paraphrase recognition algorithms are evaulated on and incorporate semantic similarity measures trained on acquired paraphrases. Therefore, we subsequently give an overview over established paraphrase acquisition approaches.

2.3 Paraphrase Acquisition

Paraphrase acquisition[5] is the process of collecting or generating phrase-paraphrase pairs, often for a given set of phrases. All strategies require a subsequent verification of the acquired paraphrases, either done by experts or trusted crowdworkers.

2.3.1 Sentential Paraphrases

Most literature on paraphrase acquisition deals with sentential or sentence-level paraphrases. Bouamor et al. (2012) identify five strategies such as the translation based methods (Zhou et al., 2006) using parallel corpora or alignment of topic-clustered news articles (Dolan and Brockett, 2005).

Via Crowdsourcing In an outstanding approach, Chen and Dolan (2011) collected paraphrases by asking crowdworkers to describe short

[5] also referred to as *paraphrase generation*

videos. A more cost-effective multi-stage crowdsourcing framework was presented by Negri et al. (2012) with the goal to increase lexical divergence of the collected paraphrases.

2.3.2 Sub-Sentential Paraphrases

Incorporating sub-sentential paraphrases in machine translation metrics also used for paraphrase detection has proven effective (Madnani et al., 2012). A large corpus consisting of more than 15 million sub-sentential paraphrases was assembled by Bannard and Callison-Burch (2005) using a pivot-based paraphrase acquisition method.

Via Crowdsourcing Buzek et al. (2010) acquired paraphrases of sentence parts problematic for translation systems using AMAZON MECHANICAL TURK. Bouamor et al. (2012) collected sub-sentential paraphrases in the context of a web-based game.

2.3.3 Passage-level paraphrases

Passage-level paraphrase acquisition has been treated within the context of the evaluation lab on uncovering plagiarism, authorship, and social software misuse (PAN) (Potthast et al., 2010): Burrows et al. (2013) acquired passage-level paraphrases for the WEBIS-CPC-11 corpus via crowdsourcing.

2.4 Semantic Similarity Evaluation

Paraphrase verification can be said to be a manual semantic similarity evaluation done by experts or trusted crowdworkers, most often on a binary scale. However, Madnani et al. (2012) believe that "binary indicators of semantic equivalence are not ideal and a continuous value [...] indicating the degree to which two pairs are paraphrastic is more suitable for most approaches". They propose averaging a large number of binary crowdworker judgements or, alternatively, a smaller number of judgements on an ordinal scale as in the SEMEVAL-2012 Semantic Textual Similarity (STS) task (Agirre et al., 2012). A continuous semantic similarity score is also used to weigh the influence of sub-sentential paraphrases used by the TERP metric.

3 Our Experiments

3.1 The CrowdFlower Platform

CROWDFLOWER is a web service for HIT providers, abstracting from the actual platform on which these tasks are run. A web interface, incorporating a graphical editor as well as the CROWDFLOWER MARKUP LANGUAGE[6] (CML), can be used to model these tasks. CROWDFLOWER provides fine-grained controls over how these tasks are executed, for instance, by restricting crowdworkers to live in specific countries or by limiting the number of HITs a single worker is allowed to complete.

Furthermore, CROWDFLOWER provides a sophisticated system to verify the correctness of the collected data, aiming at early detection and exclusion of spammers and low-accuracy workers from the job: *gold items*. Gold items consist of a HIT, e.g. a pair of paraphrases *together with* one or more possible valid answers. Once gold items are present in the dataset, workers are prompted to answer these correctly before being eligible to work on the actual data. Additionally, during the run of a job, CROWDFLOWER uses hidden gold items to revise the trustworthiness of a human worker.

3.2 Human Intelligence Task Design

Apart from gold items, the actual HIT design has the biggest impact on the quality of the collected data. Correct instructions as well as good examples have a great influence on data quality. By using CML validation features, bad user input can be prevented from being collected in the first place. Care must also be taken not to introduce an artificial bias by offering answer choices of different (time-)complexity. Within our experiments, we followed common human interface design principles such as colour coding answer options.

3.3 Crowdsourcing Sub-Sentential Paraphrase Acquisition

The biggest challenge in paraphrase acquisition via crowdsourcing is the low and varying accuracy of the crowdworkers: "The challenge [...] is automatic quality assurance; without such means the crowdsourcing paradigm is not effective, and without crowdsourcing the creation of test corpora is unacceptably expensive for realistic order of magnitudes" (Burrows et al., 2013).

We propose a new crowdsourcing strategy that allows for early detection of low-accuracy workers during the generation stage. This prevents these unwanted crowdworkers from completing

[6] CML documentation: `http://crowdflower.com/docs/cml`

HITs that would almost certainly not be validated later on. We focus on the acquisition of subsentential paraphrases for a given set of phrases, where pivot-based paraphrase acquisition methods might not be applicable. Transferring our observations to other types of paraphrases should be unproblematic.

3.3.1 Phrase-Paraphrase Generation

For this simple baseline strategy, we asked the crowdworker to generate a short phrase along with its paraphrase (p_1, p_2) while providing a small set of examples.

3.3.2 Two-Staged Paraphrase Generation

This is the traditional crowdsourcing strategy. In a first *generation* stage, we presented the crowdworker with a phrase p_1 and asked for its paraphrase p_2. In a second *validation* stage, two or three workers were asked to verify each generated phrase-paraphrase pair until an unambiguous agreement was reached. As the answers in the validation stage are binary, gold-items were added to improve the accuracy of the collected validation judgements. Negri et al. (2012) showed that after such a validation stage, expert raters agreed in 92% of the cases with the aggregated crowdworker judgements. However, the generation stage is without gold and we cannot exclude low accuracy workers early enough not to cost money. We used the regular expression verifier provided by CROWDFLOWER to ensure that the generated paraphrases contain at least one word and are not equal to the given phrases. Other than this however, the worker could enter any text.

Input Phrases As input data, we required meaningful chunks. For this, any *constituent* of a sentence can be used. A small number of examples suggested that verb phrases have a high potential of yielding interesting paraphrases, as they often have to be replaced as an isolated unit *("get a flu" → "catch a cold").* Therefore, we extracted verb phrases of two to five words from a source corpus. For this, we used the POS tagger of NLTK[7] (A Maxent Treebank POS tagger trained on Penn Treebank) and a simple chunking grammar parser.

Offering a Choice of Input Phrase A crowdworker might not always be able to come up with a paraphrase for a given phrase. If a worker receives one chunk at a time, he has to deal with it no matter how unfeasible it is for paraphrasing. One solution to this problem would be to offer a back-out option, in which a worker could declare a unit as *unsolvable* and possibly explain why. This however could easily be exploited by human workers, resulting in many unsolved items. An alternative solution is to offer workers a choice of the input phrase they want to paraphrase. We designed a HIT with a set of three different input phrases of which they have to pick one to paraphrase. If one of these options is repeatedly declined by multiple workers, we can declare it as *bad*, without having a worker pass on a unit. However, it turned out that less than 1% proved *unsolvable* and we therefore deemed such measures unnecessary.

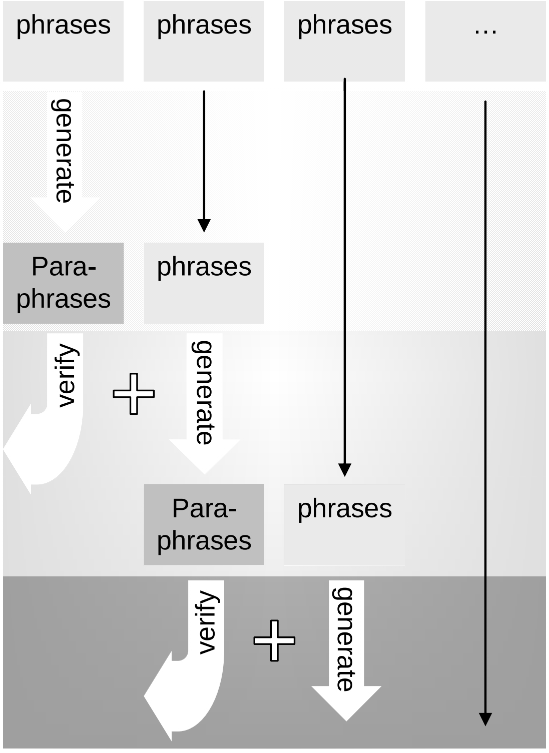


Figure 2: Illustration of the multi-stage paraphrase generation process

3.3.3 Multi-Staged Paraphrase Generation

We improved the traditional two-stage approach by combining the generation and verification steps. The task to decide whether a given pair is a paraphrase is combined with the task of paraphrasing a chunk. The matching of verification and generation items is arbitrary. Figure 2 illustrates this approach. After an initial *generate* stage, subsequent stages are combined *verify/generate* jobs. The benefit of this approach is that verification of

[7] NATURAL LANGUAGE TOOLKIT (NLTK): `http://nltk.org/`

phrase pairs allows the usage of gold-items. We can now assess the trustworthiness of a crowdworker through gold, and we indirectly *infer* their ability to *paraphrase* from their ability to decide if two items are paraphrases. The aim of this process is to reduce the number of incorrect paraphrases being generated in the first place, and thus improve the efficiency of the CROWDFLOWER task.

In contrast to Negri et al. (2012), we did not restrict access to the later stages of this job to high-accuracy workers of previous stages since our intermingled gold-items are expected to filter out low-accuracy workers in each succeeding stage. Therefore, we expect to attract contributors from a bigger pool of possibly cheaper workers.

3.3.4 Evaluation

While only 28% of the collected pairs were validated after the traditional two-staged paraphrase generation, this percentage increased to 80% in the second validation stage belonging to the multi-stage approach. Although the experiment was conducted on a small number of phrases, this result is a good indicator that our hypothesis is correct and that a combined generation and verification stage with gold items can reduce costs by early exclusion of low-accuracy workers.

Lexical divergence measures (TERP) decline, but this is expected after filtering out possibly highly divergent non-paraphrastic pairs. While our generation costs per non-validated sub-sentential paraphrase were around the same as those reported by Buzek et al. (2010) (0.024$), the costs for validated sub-sentential paraphrases were not much higher (0.06$). Negri et al. (2012) report costs of 0.27$ per sentential paraphrase, however these costs are difficult to compare, also because we did not optimize for lexical divergence.

3.4 Crowdsourcing Semantic Similarity Evaluation

We conducted an experiment in order to determine how to optimally query continuous semantic similarity scores from crowdworkers. The two different examined methods originally proposed by Madnani et al. (2012) are binary and senary[8] semantic similarity evaluation. Paraphrases were taken from the MSRPC. Optimality was defined by two different criteria: First, we analysed how well the (binary) paraphrase classification by domain experts on the MSRPC can be reproduced from our collected judgements. Second, we analysed how consistent our collected judgements are. Since we could not find any reference corpus for semantic similarity evaluation apart from the SEMEVAL-2012 STS gold that was also acquired via crowdsourcing, we resorted to training a machine learning classifier and comparing relative performance on the collected training data.

3.4.1 Binary Semantic Similarity

Crowdworkers were asked to give a binary classification of two phrases as either paraphrastic or non-paraphrastic. Binary decisions were enforced since no third option was given. Three examples of valid paraphrases were given.

A minimum of 20 judgements each for 207 phrase pairs were collected for 0.01$ per judgement. In order to deter spammers and the most inaccurate workers, we converted 14% of the phrase pairs - those with high expected inter-rater agreement - to gold items. Low inter-rater agreement on a phrase pair hinted at medium, high inter-rater agreement hinted at low or high semantic similarity. Trusted crowdworkers had an average gold accuracy of 93% on these gold items.

3.4.2 Senary Semantic Similarity

Crowdworkers were asked to give a senary classification of two phrases. The six classes were equivalent to those defined by the SemEval STS task. A short annotation guide consisting of one example per category was provided.

A minimum of 8 judgements each for 667 phrase pairs were collected for 0.02$ per judgement. In order to deter spammers and the most inaccurate workers, we converted 13% of the phrase pairs to gold items. Gold items were accepted as long as the judgement lay within an acceptable range of an expected similarity value.

3.4.3 Input Aggregation and Normalization

The following two phrase pairs demonstrate the relationship between binary inter-rater agreement and aggregated senary semantic similarity:

1. „It appears that many employers accused of workplace discrimination will be considered guilty until they can prove themselves innocent," he said.

   Employers accused of workplace discrimination now are considered guilty until they can prove themselves innocent.

[8] senary: $\{0, 1, 2, 3, 4, 5\}$ as opposed to binary $\{0, 1\}$.

Name	Stage	# Phrase Pairs	TERP
Phrase-Paraphrase Generation	Generation	100	0.89
Two-Staged Generation	1. Generation	378	0.85
	2. Validation	109 (28%)	0.68
Multi-Staged Generation	3. Generation + Gold	165	0.72
	4. Validation	134 (**80%**)	0.64

Table 1: Two-staged (1. - 2.) and multi-staged (1. - 4.) paraphrase generation results. Percentage values denote the amount of validated pairs relative to the preceding generation stage.

2. Sixteen days later, as superheated air from the shuttle's reentry rushed into the damaged wing, "there was no possibility for crew survival," the board said.

   Sixteen days later, as superheated air from the shuttle's re-entry rushed into the damaged wing, there was no possibility for crew survival, the board said.'

The binary inter-rater agreement for the first phrase pair is low (10%), so crowdworkers seemingly could not decide between paraphrastic and non-paraphrastic. Accordingly, the averaged senary semantic similarity takes an intermediate value (3.4).

The binary inter-rater agreement for the second phrase pair however is very high (100%), so we expect the sentences to be either clearly non-paraphrastic or clearly paraphrastic. A maximal averaged senary semantic similarity value of 5.0 confirms this intuition.

In order to make aggregated binary and senary input comparable, we scaled the binary judgements so that the sampled average and variance matched that of the senary judgements. These semantic similarities are strongly correlated (3a) with Pearson coefficient of 0.81 and seem to respect the MSRPC expert annotator rating with positive correlation between aggregated semantic similarity and binary MSRPC classification.

With reference to Denkowski and Lavie (2010), we used the following aggregation and normalization techniques:

Straight Average The aggregated semantic similarity is the average of all collected judgements. This is our baseline approach.

Judge Normalization To compensate for different evaluation standards, each judge's judgements are scaled so that its sample average and variance matches that of the average (3b).

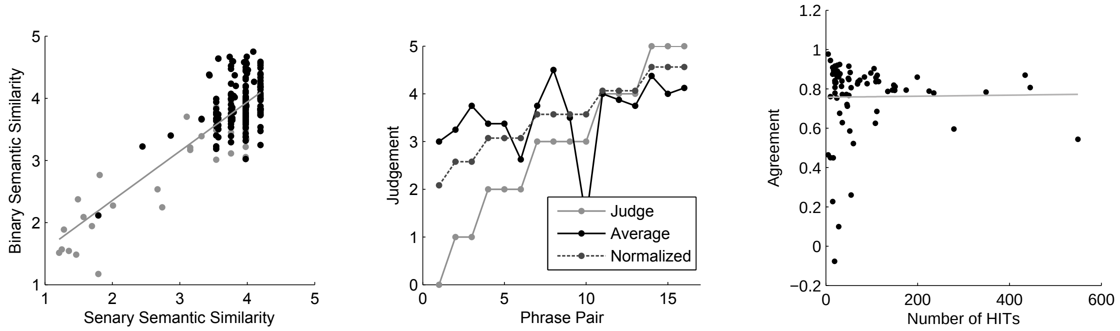
Judge Outlier Removal Removing judges whose inter-rater agreement with the average is less than 0.5; motivated by Agirre et al. (2012): "Given the high quality of the annotations among the turkers, we could alternatively use the correlation between the turkers itself to detect poor quality annotators".

Weighted Voting Each judge's judgements are weighted by its inter-rater agreement with the average.

We also wanted to know whether limiting the amount of possible HITs or judgements per crowdworker could increase the quality of the collected judgements. However, while high-throughput crowdworkers showed lower variance in their agreement compared to crowdworkers with a small number of completed HITs, correlation between the number of completed HITs and agreement was very weak (3c) with Pearson coefficient of 0.01.

3.4.4 Machine Learning Evaluation

We trained the UKP machine learning classifier originally developed for the Semantic Textual Similarity (STS) task at SemEval-2012 (Bär et al., 2012) on the averaged binary and senary judgements for 207 identical phrase pairs. Since we were not interested in the performance of the machine learning classifier but in the quality of the collected data, we measured the relative performance of the learned model on the training data. The number of training examples remained constant. This was repeated multiple times while varying the number of judgements used in the aggregation of the semantic similarity values. We observed that with increasing number of judgements, the correlation coefficient converges seemingly against an upper bound (binary: 0.68 for 20 judgements, senary: 0.741 for 8 judgements). The



(a) Correlation between aggregated senary and binary semantic similarity (black = paraphrases according to MSRPC)

(b) Judge normalization

(c) The activity of a crowdworker does not correlate with agreement to the average

Figure 3: Input aggregation and normalization

machine learning classifier performs best when trained on semantic similarity data collected on a senary scale (4). Even if we only take the first three senary judgements per phrase pair into account, it is still superior to 20 binary judgements although the total amount of information queried from the crowdworkers is much smaller.

In a second step, we compared the performance while employing different input normalization techniques on the whole set of 667 phrase pairs with senary judgements. While all techniques increased the trained classifier's performance, weighted voting performed best (2).

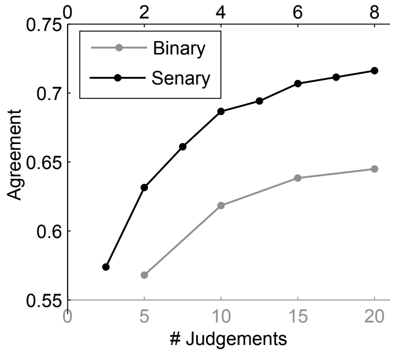


Figure 4: Machine learning results (agreement = correlation with training data)

Technique	Correlation
Straight Average	0.716
Judge Outlier Removal	0.719
Judge Normalization	0.721
Weighted Voting	**0.722**

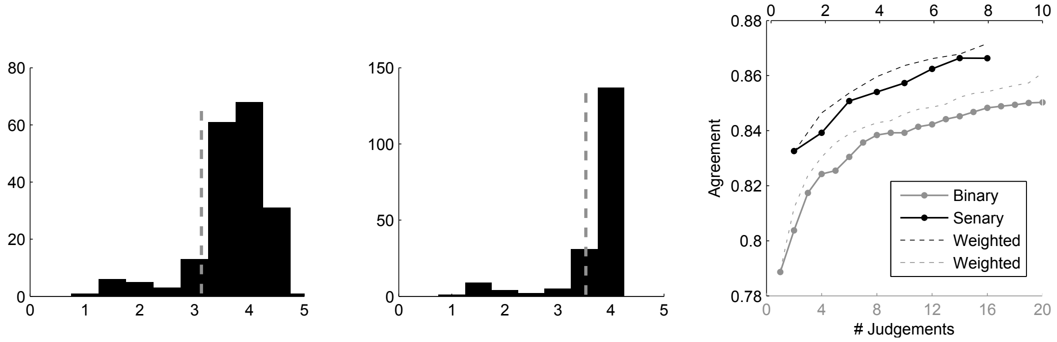
Table 2: Input normalization results

3.4.5 MSRPC Evaluation

In addition to the machine learning evaluation, we compared our results to the binary semantic similarity classification given by the MSRPC expert annotators. In order to do so, we had to find an optimal threshold in $[0, 5]$ splitting our semantic similarity range in two, dividing paraphrastic from non-paraphrastic phrase pairs. Again, this was repeated multiple times while varying the number of judgements used in the aggregation of the semantic similarity values. However, this time we did not simply take the first n judgements each, but averaged over different possible sampling combinations. We measured percentage agreement with MSRPC and the optimal threshold for non-weighted and weighted judgements, since weighted voting performed best in the machine learning evaluation (5c).

Surprisingly, even for binary paraphrastic-non-paraphrastic classification, querying a senary semantic similarity value from crowdworkers yields better results than directly asking for a binary classification. However, the results also indicate that in both cases, input normalization plays an important role and agreement could be improved by more sophisticated or combined input normalization techniques as well as by collecting additional judgements.

A semantic similarity of 3.1 (senary) (5a) respectively 3.5 (binary) (5b) corresponds optimally to the paraphrastic-non-paraphrastic threshold chosen by the MSRPC expert annotators. Costs per evaluated phrase pair were at 0.16$



(a) Optimal threshold for senary semantic similarity is 3.1

(b) Optimal threshold for binary semantic similarity is 3.5

(c) Average agreement with MSRPC with optimal threshold per number of judgements

Figure 5: MSRPC evaluation (agreement = percentual agreement with aggregated judgements)

(senary, 8 judgements) compared to 0.20$ for the SEMEVAL-2012 STS task (senary, 5 judgements). However, we did not examine how this and possible further cost reduction impacts agreement with MSRPC.

4 Conclusion

We presented a multi-stage crowdsourcing approach tackling the problem of missing gold in paraphrase generation. This approach has shown to work very well for sub-sentential paraphrase generation and we strongly believe that it will work equally well for sentential paraphrase generation, resulting in significantly reduced costs of paraphrase corpus creation.

We also compared different crowdsourcing approaches towards semantic similarity evaluation, showing that for both semantic similarity evaluation on a continuous and a binary scale, querying an ordinal senary semantic similarity value from crowdworkers yields better results than directly asking for a binary classification.

5 Future Work

Our goal to sub-sentential paraphrase generation was cost minimization by early removal of low-accuracy workers. Apart from being grammatical and paraphrastic, we did not enforce other quality constraints on the collected data. A combination of our multi-stage approach with that of Negri et al. (2012) could prove successful if both cost and quality, i.e. lexical divergence between phrase-paraphrase pairs, are to be optimized.

There is also room for reducing the cost of the verification stage e.g. by automatically filtering out paraphrases before presenting them to a crowdworker using e.g. lexical divergence, length of the sentence or other measures as it was done by Burrows et al. (2013).

Another interesting question we could not answer due to budget constraints is: Can the crowd replace the expert and if yes, how many crowdworkers are needed to do so reliably? One possible way to answer this question for paraphrase evaluation would be to collect semantic similarity judgements for the whole MSRPC and to see how many judgements per phrase are needed to reliably reproduce the MSRPC classification results with an inter-rater agreement of 84% for the whole corpus.

Acknowledgements

The authors would like to thank Chris Biemann of TU Darmstadt, Germany, for pointing us to the problem of paraphrase evaluation via crowdsourcing leading to this research as well as his supervision and helpful suggestions. We also thank our reviewers for their feedback.

References

Eneko Agirre, Daniel Cer, Mona Diab, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 385–393, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

Colin Bannard and Chris Callison-Burch. 2005. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 597–604, Stroudsburg, PA, USA. Association for Computational Linguistics.

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation, held in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada, Jun.

Houda Bouamor, Aurélien Max, Gabriel Illouz, and Anne Vilnat. 2012. A contrastive review of paraphrase acquisition techniques. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

Ram Boukobza and Ari Rappoport. 2009. Multi-word expression identification using sentence surface features. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 468–477, Singapore, August. Association for Computational Linguistics.

Steven Burrows, Martin Potthast, and Benno Stein. 2013. Paraphrase Acquisition via Crowdsourcing and Machine Learning. *Transactions on Intelligent Systems and Technology (ACM TIST) (to appear)*.

Olivia Buzek, Philip Resnik, and Benjamin B. Bederson. 2010. Error driven paraphrase annotation using mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 217–221, Stroudsburg, PA, USA. Association for Computational Linguistics.

David L. Chen and William B. Dolan. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 190–200, Portland, Oregon, USA, June.

Michael Denkowski and Alon Lavie. 2010. Exploring normalization techniques for human judgments of machine translation adequacy collected using amazon mechanical turk. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, CSLDAMT '10, pages 57–61, Stroudsburg, PA, USA. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Third International Workshop on Paraphrasing (IWP2005)*. Asia Federation of Natural Language Processing.

Mark Dras. 1999. *Tree Adjoining Grammar and the Reluctant Paraphrasing of Text*. Ph.D. thesis, Macquarie University.

Christiane Fellbaum, Alexander Geyken, Axel Herold, Fabian Koerner, and Gerald Neumann. 2006. Corpus-based Studies of German Idioms and Light Verbs. *International Journal of Lexicography*, 19(4):349–360, December.

Samuel Fernando and Mark Stevenson. 2008. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics*.

Dekang Lin and Patrick Pantel. 2001. Discovery of inference rules for question-answering. *Nat. Lang. Eng.*, 7(4):343–360, December.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 182–190, Stroudsburg, PA, USA. Association for Computational Linguistics.

Erwin Marsi and Emiel Krahmer. 2010. Automatic analysis of semantic similarity in comparable text through syntactic tree matching. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 752–760, Beijing, China, August. Coling 2010 Organizing Committee.

Matteo Negri, Yashar Mehdad, Alessandro Marchetti, Danilo Giampiccolo, and Luisa Bentivogli. 2012. Chinese whispers: Cooperative paraphrase acquisition. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may.

Martin Potthast, Alberto Barrón-Cedeño, Andreas Eiselt, Benno Stein, and Paolo Rosso. 2010. Overview of the 2nd international competition on plagiarism detection. *Notebook Papers of CLEF*, 10.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Ter-plus: Paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2):117–127.

Richard Socher, Eric H. Huang, Jeffrey Pennington, Andrew Y. Ng, and Christopher D. Manning. 2011. Dynamic Pooling and Unfolding Recursive Autoencoders for Paraphrase Detection. In *Advances in Neural Information Processing Systems 24*.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. 2006. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, pages 77–84, Stroudsburg, PA, USA. Association for Computational Linguistics.