Identifying Problems and Solutions in Scientific Text

Kevin Heffernan and Simone Teufel

University of Cambridge
Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, UK
forname.surname@cl.cam.ac.uk

Abstract. Research is often described as a problem-solving activity, and as a result, descriptions of problems and solutions are an essential part of the scientific discourse used to describe research activity. We present an automatic classifier that, given a phrase that may or may not be a description of a scientific problem or a solution, makes a binary decision about problemhood and solutionhood of that phrase. We recast the problem as a supervised machine learning problem, and define a set of 8 features correlated with the target categories, and use several machine learning algorithms on this task. We also create our own corpus of 2000 positive and negative examples of problems and solutions. We find that we can distinguish problems from non-problems with an accuracy of 82.4%, and solutions from non-solutions with an accuracy of 81.5%. Our two most helpful features for the task are syntactic information (POS tags) and document and word embeddings.

1 Introduction

Problem solving is generally regarded as the most important cognitive activity in everyday and professional contexts [1]. Many studies on formalising the cognitive process behind problem-solving exist, for instance [2]. [3] argues that we all share knowledge of the thought/action problem-solution process involved in real life, and so our writings will often reflect this order. There is general agreement amongst theorists that state that the nature of the research process can be viewed as a problem-solving activity [4–7].

One of the best-documented problem-solving patterns was established by Winter [8]. Winter analysed thousands of examples of technical texts, and noted that these texts can largely be described in terms of a four-part pattern consisting of Situation, Problem, Solution and Evaluation. This is very similar to the pattern described by [5], which consists of Introduction-Theory, Problem-Experiment-Comment and Conclusion. The difference is that in Winter's view, a solution only becomes a solution after it has been evaluated positively. Hoey changes Winter's pattern by introducing the concept of Response in place of Solution [9]. This seems to describe the situation in science better, where evaluation is mandatory for research solutions to be accepted by the community. In

Hoey's pattern, the Situation (which is generally treated as optional) provides background information; the Problem describes an issue which requires attention; the Response provides a way to deal with the issue, and the Evaluation assesses how effective the response is.

An example of this pattern in the context of the Goldilocks story can be seen in Figure 1. In this text, there is a preamble providing the setting of the story (i.e. Goldilocks is lost in the woods), which is called the Situation in Hoey's system. A Problem in encountered when Goldilocks becomes hungry. Her first Response is to try the porridge in big bear's bowl, but she gives this a negative Evaluation ("too hot!") and so the pattern returns to the Problem. This continues in a cyclic fashion until the Problem is finally resolved by Goldilocks giving a particular Response a positive Evaluation of baby bear's porridge ("it's just right").

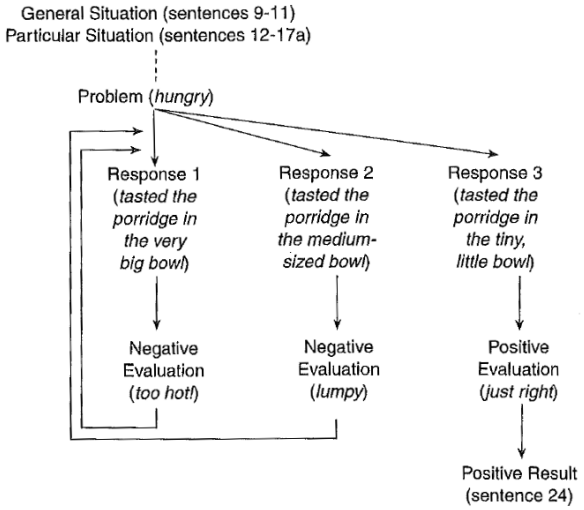

Fig. 1: Example of problem-solving pattern when applied to the Goldilocks story, from [9].

It would be attractive to detect problem and solution statements automatically in text. This holds true both from a theoretical and a practical viewpoint. Theoretically, we know that sentiment detection is related to problem-solving activity, because of the perception that "bad" situations are transformed into "better" ones via problem-solving. The exact mechanism of how this can be detected would advance the state of the art in text understanding. In terms of linguistic realisation, problem and solution statements come in many variants and reformulations, often in the form of positive or negated statements about the conditions, results and causes of problem–solution pairs. Detecting and interpreting those would give us a reasonably objective manner to test a system's understanding capacity. Practically, being able to detect any mention of a prob-

lem is a first step towards detecting a paper's specific research goal. Being able to do this has been a goal for scientific information retrieval for some time, and if successful, it would improve the effectiveness of scientific search immensely. Detecting problem and solution statements of papers would also enable us to compare similar papers and eventually even lead to automatic generation of review articles in a field.

There has been some computational effort on the task of identifying problem-solving patterns in text. However, most of the prior work has not gone beyond the usage of keyword analysis and some simple contextual examination of the pattern. [10] presents a corpus-based analysis of lexio-grammatical patterns for problem and solution clauses using articles from professional and student reports. Problem and solution keywords were used to search their corpora, and each occurrence was analysed to determine grammatical usage of the keyword. More interestingly, the causal category associated with each keyword in their context was also analysed. For example, Reason-Result or Means-Purpose were common causal categories found to be associated with problem keywords.

The goal of the work by [11] was to determine words which are semantically similar to problem and solution, and to determine how these words are used to signal problem-solution patterns. However, their corpus-based analysis used articles from the Guardian newspaper. Since the domain of newspaper text is very different from that of scientific text, we decided not to consider those keywords associated with problem-solving patterns for use in our work.

Instead of a keyword-based approach, [12] used discourse markers to examine how the problem-solution pattern was signalled in text. In particular, they examined how adverbials associated with a result such as *"thus, therefore, then, hence"* are used to signal a problem-solving pattern. Problem solving also has been studied in the framework of discourse theories such as Rhetorical Structure Theory [13] and Argumentative Zoning [14]. RST uses a solutionhood relation as one of the 23 relations that can hold between elementary discourse units. However, the definition of problemhood used in RST differs too much from ours to be of direct use here. Argumentative Zoning [14] contains zones such as *Gap/Weak* which have a close relation to our definition of problemhood, and so knowledge from this particular zone may prove beneficial in the future, although we do not study this effect in the current paper.

In this work, we approach the task of identifying problem-solving patterns in scientific text. We choose to use the model of problem-solving described by Hoey [9]. This pattern comprises four parts: Situation, Problem, Response and Evaluation. The Situation element is considered optional to the pattern, and so our focus centres on the core pattern elements.

2 Goal statement and task

Many surface features in the text offer themselves up as potential signals for detecting problem-solving patterns in text. However, since Situation is an optional element, we decided to focus on either Problem or Response and Evaluation as

signals of the pattern. Moreover, we decide to look for each type in isolation. Our reasons for this are as follows: It is quite rare for an author to introduce a problem without resolving it using some sort of response, and so this is a good starting point in identifying the pattern. There are exceptions to this, as authors will sometimes introduce a problem and then leave it to future work, but overall there should be enough signal in the Problem element to make our method of looking for it in isolation worthwhile. The second signal we look for is the use of Response and Evaluation within the same sentence. Similar to Problem elements, we hypothesise that this formulation is well enough signalled externally to help us in detecting the pattern. For example, consider the following Response and Evaluation: "One solution is to use smoothing." In this statement, the author is explicitly stating that smoothing is a solution to a problem which must have been mentioned in a prior statement. In scientific text, we often observe that solutions implicitly contain both Response and Evaluation (positive) elements. Therefore, due to these reasons there should be sufficient external signals for the two pattern elements we concentrate on here.

When attempting to find Problem elements in text, we run into the issue that the word "problem" actually has at least two word senses that need to be distinguished. There is a word sense of "problem" that means something which must be undertaken (i.e. task), while another sense is the core sense of the word, something that is problematic and negative. Only the latter sense is aligned with our sense of problemhood. This is because the simple description of a task does not predispose problemhood, just a wish to perform some act. Consider the following examples, where the non-desired word sense is being used:

- *"Das and Petrov (2011) also consider the problem of unsupervised bilingual POS induction."* [15].
- *"In this paper, we describe advances on the problem of NER in Arabic Wikipedia."* [16].

Here, the author explicitly states that the phrases in orange are problems, they align with our definition of research tasks and not with what we call here 'problematic problems'. We will now give some examples from our corpus for the desired, core word sense:

- *"The major limitation of supervised approaches is that they require annotations for example sentences."* [17].
- *"To solve the problem of high dimensionality we use clustering to group the words present in the corpus into much smaller number of clusters."* [18].

When creating our corpus of positive and negative examples, we took care to select only problem strings that satisfy our definition of problemhood; section 3 will explain how we did that.

3 Corpus creation

Our new corpus is a subset of the latest version of the ACL anthology released in March, 2016[1] which contains 22,878 articles in the form of PDFs and OCRed text.[2] The 2016 version was also parsed using ParsCit [19]. ParsCit recognises not only document structure, but also bibliography lists as well as references within running text. A random subset of 2,500 papers was collected covering the entire ACL timeline. In order to disregard non-article publications such as introductions to conference proceedings or letters to the editor, only documents containing abstracts were considered. The corpus was preprocessed using tokenisation, lemmatisation and dependency parsing with the Rasp Parser [20].

Our goal was to define a ground truth for problem and solution strings, while covering as wide a range as possible of syntactic variations in which such strings naturally occur. However, to simplify the task, we only consider here problem and solution descriptions that are at most one sentence long. In reality, of course, many problem descriptions and solution descriptions go beyond single sentence, and require for instance an entire paragraph. However, we also know that short summaries of problems and solutions are very prevalent in science, and also that these tend to occur in the most prominent places in a paper. This is because scientists are trained to express their contribution and the obstacles possibly hindering their success, in an informative, succinct manner. That is the reason why we can afford to only look for shorter problem and solution descriptions, ignoring those that cross sentence boundaries.

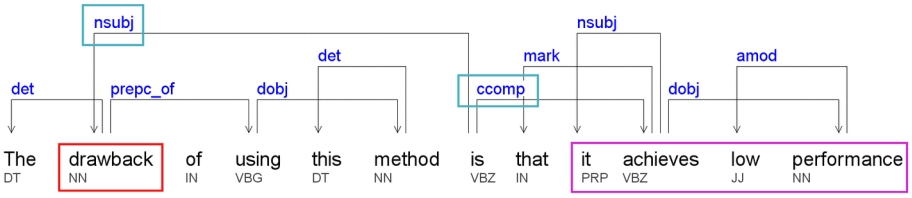

Fig. 2: Example of our extraction method for problems using dependencies.

To define our ground truth, we examined the parsed dependencies and looked for a target word ("problem/solution") in subject position, and then chose its syntactic argument as our candidate problem or solution phrase. To increase the variation, i.e., to find as many different-worded problem and solution descriptions as possible, we additionally used semantically similar words (near-synonyms) of the target words "problem" or "solution" for the search. Semantic similarity

[1] http://acl-arc.comp.nus.edu.sg/

[2] The corpus comprises 3,391,198 sentences, 71,149,169 words and 451,996,332 characters.

was defined as cosine in a deep learning distributional vector space, trained using Word2Vec [21] on 18,753,472 sentences from a biomedical corpus based on all full-text Pubmed articles [22]. From the 200 words which were semantically closest to "problem", we manually selected 28 clear synonyms. From the 200 semantically closest words to "solution" we similarly chose 19. Of the sentences matching our dependency search, a subset of problem and solution candidate sentences were randomly selected. An example of this is shown in Figure 2. Here, the target word "drawback" is in subject position (highlighted in red), and its clausal argument (ccomp) is "(that) it achieves low performance" (highlighted in purple). Examples of other arguments we searched for included copula constructions and direct/indirect objects.

If more than one candidate was found in a sentence, one was chosen at random. Non-grammatical sentences were excluded; these might appear in the corpus as a result of its source being OCRed text.

The potential phrases expressing problems and solutions, respectively, were then independently checked for correctness by two annotators (the two authors of this paper). Correctness was defined by two criteria:

- The sentence must unambiguously and clearly state the phrase's status as either a problem or a solution. For problems, the guidelines state that the phrase has to represent one of the following:

  1. an unexplained phenomenon or a problematic state in science; or
  2. a research question; or
  3. an artifact that does not fulfil its stated specification.

  For solutions, the phrase had to represent a response to a problem with a positive evaluation. Implicit solutions were also allowed.

- The phrase must not lexically give away its status as problem or solution phrase.

The second criterion saves us from machine learning cues that are too obvious. If for instance, the phrase itself contained the words "lack of" or "problematic" or "drawback", our manual check rejected it, because it would be too easy for the machine learner to learn such cues, at the expense of many other, more generally occurring cues.

We next needed to find negative examples for both cases. We wanted them not to stand out on the surface as negative examples, so we chose them so as to mimic the obvious characteristics of the positive examples as closely as possible. We call the negative examples 'non-problems' and 'non-solutions' respectively. We wanted the only differences between problems and non-problems to be of a semantic nature, nothing that could be read off on the surface. We therefore sampled a population of phrases that obey the same statistical distribution as our problem and solution strings while making sure they really are negative examples. We started from sentences *not* containing any problem/solution words

(i.e. those used as target words). From each such sentence, we at random selected one syntactic subtree contained in it. From these, we randomly selected a subset of negative examples of problems and solutions that satisfy the following conditions:

- The distribution of the head POS tags of the negative strings should perfectly match the head POS tags[3] of the positive strings. This has the purpose of achieving the same proportion of surface syntactic constructions as observed in the positive cases.

- The average lengths of the negative strings must be within a tolerance of the average length of their respective positive candidates e.g., non-solutions must have an average length very similar (i.e. +/- small tolerance) to solutions. We chose a tolerance value of 3 characters.

Again, a human quality check was performed on non-problems and non-solutions. For each candidate non-problem statement, the candidate was accepted if it did not contain a phenomenon, a problematic state, a research question or a non-functioning artefact. If the string expressed a research task, without explicit statement that there was anything problematic about it (i.e., the 'wrong' sense of "problem", as described above), it was allowed as a non-problem. A clause was confirmed as a non-solution if the string did not represent both a response and positive evaluation.

If the annotator found that the sentence had been slightly mis-parsed, but did contain a candidate, they were allowed to move the boundaries for the candidate clause. This resulted in cleaner text, e.g., in the frequent case of coordination, when non-relevant constituents could be removed.

From the set of sentences which passed the quality-test for both independent assessors, 500 instances of positive and negative problems/solutions were randomly chosen (i.e. 2000 instances in total).

4 Method

4.1 Experimental design

In our experiments, we used three classifiers, namely Naïve Bayes, Logistic Regression and a Support Vector Machine. For all classifiers an implementation from the WEKA machine learning library [24] was chosen. Given that our dataset is small, 10-fold cross-validation was used instead of a held out test set. All significance tests were conducted using the (two-tailed) Sign Test [25].

[3] The head POS tags were found using a modification of the Collins' Head Finder. This modified algorithm addresses some of the limitations of the head finding heuristics described by [23] and can be found here: http://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/trees/ModCollinsHeadFinder.html.

4.2 Linguistic correlates of problem- and solution-hood

We first define a set of features without taking the phrase's context into account. This will tell us about the disambiguation ability of the problem/solution description's semantics alone. In particular, we cut out the rest of the sentence other than the phrase and never use it for classification. This is done for similar reasons to excluding certain 'give-away' phrases *inside* the phrases themselves (as explained above). As the phrases were found using templates, we know that the machine learner would simply pick up on the semantics of the template, which always contains a synonym of "problem" or "solution", thus drowning out the more hidden features hopefully inherent in the semantics of the phrases themselves. If we allowed the machine learner to use these stronger features, it would suffer in its ability to generalise to the real task.

Of these, bags of words are traditionally successfully used for classification tasks in NLP, so we included bags of words (lemmas) within the candidate phrases as one of our features (and treat it as a baseline later on). Our second feature concerns the polarity of selected words, by determining the head of each candidate phrase and performing word sense disambiguation of each head using the Lesk algorithm [26]. The polarity of the resulting synset in SentiWordNet [27] was then looked up and used as a feature. Next, a set of syntactic features were defined by using the POS tags present in each candidate. We were careful not to base the model directly on the head POS tag and the length of each candidate phrase, as these are defining characteristics used for determining the non-problem and non-solution candidate set. Not all words are assigned a sense by the Lesk algorithm, so we need to take care when that happens to a phrasal head. In those cases, the distributional semantic similarity of the phrasal head is compared to two words with a known polarity, namely "poor" and "excellent".These particular words have traditionally been consistently good indicators of polarity status in many studies [28, 29]. Semantic similarity was defined as cosine similarity on the embeddings of the Word2Vec model (cf. Section 3).

Given that solutions often involve an activity (e.g. a task), we also model the subcategorisation properties of the verbs involved. Our intuition was that since problematic situations are often described as non-actions, then these are more likely to be intransitive. Conversely solutions are often actions and are likely to have at least one argument. This feature was calculated by running the C&C parser [30] on each sentence. C&C is a supertagger and parser that has access to subcategorisation information.

We also wanted to add more information using word embeddings. This was done in two different ways. Firstly, we created a Doc2Vec model [31], which was trained on ∼19 million sentences from scientific text (no overlap with our data set). An embedding was created for each candidate sentence. Secondly, word embeddings were calculated using the Word2Vec model (cf. Section 3). For each candidate head, the full word embedding was included as a feature.

Modality Responses to problems in scientific writing often express possibility and necessity, and so have a close connection with modality. Modality can be

broken into three main categories, as described by [32], namely epistemic (possibility), deontic (permission / request / wish) and dynamic (expressing ability).

Problems have a strong relationship to modality within scientific writing. Often, this is due to a tactic called "hedging" [33] where the author uses speculative language in an attempt to make either noncommital or vague statements. This has the effect of allowing the author to distance themselves from the statement, and is often employed when discussing negative or problematic topics.

To take this linguistic correlate into account as a feature, we replicated a modality classifier as described by Ruppenhofer et al. [34]. More sophisticated modality classifiers have been recently introduced, for instance using a wide range of features and convolutional neural networks, e.g, [35, 36]. However, we wanted to check the effect of a simpler method of modality classification on the final outcome first before investing heavily into their implementation. We trained three classifiers using the subset of features which Ruppenhofer et al. reported as performing best, and evaluated them on the gold standard dataset provided by the authors[4]. The dataset contains annotations of English modal verbs on the 535 documents of the first MPQA corpus release [37].

The modality classifier was then retrained on the entirety of the dataset used by [34] using the best performing model from training (Logistic Regression). This new model was then used in the upcoming experiment to predict modality labels for each instance in our dataset.

5 Results

5.1 Problems

Feature Sets	**Classification Accuracy**		
	NB	**LR**	**SVM**
$\text{Baseline}_{\text{bow}}$	62.7	67.9	68.1
+Transitivity	63.1	68.3	68.1
+Modality	63.9	67.2	68.2
+Polarity	63.6	65.4	68.4
+Syntax	71.8*	72.5*	74.6
+Doc2Vec	80.8**	79.9*	79.6
+Word2Vec	82.4	81.0	80.6
$+\text{Word2Vec}_{\text{smoothed}}$	82.2	81.1	80.9

Fig. 3: Results distinguishing problems from non-problems using Naïve Bayes (NB), Logistic Regression (LR) and a Support Vector Machine (SVM). Each consecutive feature set is cumulative. 10-fold stratified cross-validation was used across all experiments. Statistical significance with respect to the previous feature set at the $p < 0.05$ and 0.01 levels is denoted by * and ** respectively.

[4] https://www.uni-hildesheim.de/ruppenhofer/data/modalia_release1.0.tgz

IG	Features
0.018	single
0.014	limit, experiment
0.010	datum, information
0.009	require, generate, error, many
0.008	explosion

Fig. 4: Information gain (IG) in bits of top lemmas from the best performing model in Figure 3 (Word2Vec with NB).

As can be seen from Figure 3, we are able to achieve good results for distinguishing a problematic statement from non-problematic one. The bag of words baseline achieves a reasonable performance of 68.1% for the SVM classifier, showing that there is enough signal in the candidate phrases alone to distinguish them much better than random chance. Taking a look at Figure 4, which shows the information gain for the top lemmas, we can see that the top lemmas are indeed indicative of problemhood (e.g. "limit","explosion"). The transitivity feature provided some improvement over the baseline but was unable to achieve the marked improvement we were expecting. Upon taking a closer look at our data, we saw that our hypothesis that intransitive verbs are commonly used in problematic statements was true, with over 30% of our problems (153) using them. However, due to our sampling method for the negative cases we also picked up many intransitive verbs (163). This explains the small improvement given that the distribution of intransitive verbs amongst the positive and negative candidates was almost even.

The modality feature was the most expensive to produce, and managed to increase performance in both the Bayesian and SVM classifiers but degraded performance in the Logistic Regression. This surprising result may be partly due to a data sparsity issue where only a small portion (67) of our instances contained modal verbs. If the accumulation of additional data was possible, we think that this feature may have the potential to be much more valuable in determining problemhood. Additionally, modality has also shown to be helpful in determining contextual polarity [38] and argumentation [39], so using the output from this modality classifier may also prove useful for further feature engineering taking this into account in future work.

Polarity also didn't achieve the performance we were expecting, improving only the SVM, but this feature also suffers from a sparsity issue resulting from cases where the Lesk algorithm [26] is not able to resolve the synset of the syntactic head. Knowledge of syntax provides a big improvement with a significant increase in results from two of the classifiers. Examining this in greater detail, POS tags with high information gain included WH- tags and VB- tags. WH-tags encode a problematic nature, that of being unsure / awaiting resolution and so this may be one of the reasons for the marked increase.

The embeddings from Doc2Vec allowed us to obtain the most significant increase in performance (80.8 with Naïve Bayes) and Word2Vec provided the

best overall result (82.4 with Naïve Bayes). The addition of these vectors may be seen as a form of smoothing in cases where previous linguistic features had a sparsity issue i.e., instead of a NULL entry, the embeddings provide some sort of value for each candidate. Particularly wrt. the polarity feature, cases where Lesk was unable to resolve a synset meant that a ZERO entry was added to the vector supplied to the machine learner. However, using the word embeddings for the head in addition to the head's polarity meant that even if Lesk was unable to resolve the synset, the embedding were able to provide some sort of signal.

5.2 Solutions

Feature Sets	**Classification Acc.**		
	NB	**LR**	**SVM**
$\text{Baseline}_{\text{bow}}$	66.8	63.4	69.5
+Transitivity	66.1	63.3	69.1
+Polarity	68.0	66.3	71.5
+Syntax	70.4	69.5	73.9
+Doc2Vec	75.5	75.5	78.8
+Word2Vec	75.0	81.5	79.8
$+\text{Word2Vec}_{\text{smoothed}}$	74.6	81.3	80.1

Fig. 5: Results distinguishing solutions from non-solutions using Naïve Bayes (NB), Logistic Regression (LR) and a Support Vector Machine (SVM). Each consecutive feature set is cumulative. 10-fold stratified cross-validation was used across all experiments.

IG	**Features**
0.022	use
0.014	method
0.013	argument
0.012	dependency
0.011	sequence, configuration, subject
0.009	weakest, label
0.008	rewrite, employ, edge
0.007	verb, lexical, feature

Fig. 6: Information gain (IG) in bits of top lemmas from the best performing model in Figure 5 (Word2Vec with LR).

The results for disambiguation of solutions from non-solutions can be seen in Figure 5. The bag of words baseline performs much better than random, with the performance being quite high with regard to the SVM (this result was also higher than any of the baseline performances from the problem classifiers). As shown in Figure 6, the top ranked lemmas from the best performing model (using information gain) were "use" and "method". These lemmas are very indicative of solutionhood and so give some insight into the high baseline returned from the machine learners. Transitivity provided no improvement and actually degraded performance in all three classifiers. However, this low performance is due to the sampling of the non-solutions (the same reason for the low performance of the problem transitivity feature). When fitting the POS-tag distribution for the negative samples, we noticed that over 80% of the head POS-tags were verbs (much higher than the problem heads). The most frequent verb type being the infinite form. This is not surprising given that a very common formulation to

describe a solution is to use the infinitive "TO" since it often describes a task e.g., *"One solution is to find the singletons and remove them."* Therefore, since the head POS tags of the non-solutions had to match this high distribution of infinitive verbs present in the solution, the transitivity feature is not particularly discriminatory. Polarity and syntactic features were slightly more discriminate, improving results in all three classifiers. However, similar to the problem experiment, the embeddings from Word2Vec and Doc2Vec provide the highest result (81.5% from LR).

There was no significant increase in performance as each feature set was added. However, the best performing models for each classifier (Doc2Vec, Word2Vec and $\text{Word2Vec}_{\text{smoothed}}$) were all significant with regard to the baseline ($P < 0.01$).

6 Discussion

In this work, we have presented new supervised classifiers for the task of identifying problem and solution statements in scientific text. We have also introduced a new corpus for this task and used it for evaluating our classifiers. Great care was taken in constructing the corpus by ensuring that the negative and positive samples were closely matched in terms of syntactic shape. If we had simply selected random subtrees for negative samples without regard for any syntactic similarity with our positive samples, the machine learner may have found easy signals such as sentence length. Additionally, since we did not allow the machine learner to see the surroundings of the candidate string within the sentence, this made our task even harder. Our performance on the corpus shows promise for this task, and proves that there are strong signals for determining both the problem and solution parts of the problem-solving pattern independently.

With regard to classifying problems from non-problems, features such as the POS tag and document and word embeddings provide the best results, with the Word2Vec embeddings providing the highest performance (82.4%). Classifying solutions from non-solutions also performs well using these features, with the best result coming from the embeddings (81.5%).

In future work, we plan to link problem and solution statements which were found independently during our corpus creation. Given that our classifiers were trained on data solely from the ACL anthology, we also hope to investigate the domain specificity of our classifiers and see how well they can generalise to domains other than ACL (e.g. bioinformatics). Since we took great care at removing the knowledge our classifiers have of the explicit statements of problem and solution (i.e. the classifiers were trained only on the syntactic argument of the explicit statement of problem-/solution-hood), our classifiers should in principle be in a good position to generalise, i.e., find implicit statements too. In future work, we will measure to which degree this is the case.

To facilitate further research on this topic, all code and data used in our experiments can be found here: www.cl.cam.ac.uk/∼kh562/ps.html

References

1. David H Jonassen. Toward a design theory of problem solving. *Educational Technology Research and Development*, 48(4):63–85, 2000.
2. Balakrishnan Chandrasekaran. Towards a taxonomy of problem solving types. *AI Magazine*, 4(1):9, 1983.
3. Michael P Jordan. Short texts to explain problem-solution structures-and vice versa. *Instructional Science*, 9(3):221–252, 1980.
4. Jörg Strübing. Research as pragmatic problem-solving: The pragmatist roots of empirically-grounded theorizing. *The Sage Handbook of Grounded Theory*, pages 580–602, 2007.
5. Teun Adrianus Van Dijk. *Text and context explorations in the semantics and pragmatics of discourse*. Longman Linguistics Library, 1980.
6. John Hutchins. On the structure of scientific texts. *UEA Papers in Linguistics*, 5(3):18–39, 1977.
7. Joseph Evans Grimes. *The thread of discourse*, volume 207. Walter de Gruyter, 1975.
8. EO Winter. *Some aspects of cohesion*. University of London, 1968.
9. Michael Hoey. *Textual interaction: An introduction to written discourse analysis*. Psychology Press, 2001.
10. Lynne Flowerdew. *Corpus-based analyses of the problem-solution pattern: a phraseological approach*, volume 29. John Benjamins Publishing, 2008.
11. Mike Scott. *Mapping key words to problem and solution*. John Benjamins Publishing, 2001.
12. Maggie Charles. Adverbials of result: Phraseology and functions in the problem–solution pattern. *Journal of English for Academic Purposes*, 10(1):47–60, 2011.
13. William C Mann and Sandra A Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281, 1988.
14. Simone Teufel. *Argumentative zoning: Information extraction from scientific text*. PhD thesis, University of Edinburgh, 2000.
15. Desai Chen, Chris Dyer, Shay B Cohen, and Noah A Smith. Unsupervised bilingual pos tagging with markov random fields. In *Proceedings of the First Workshop on Unsupervised Learning in NLP*, pages 64–71. Association for Computational Linguistics, 2011.
16. Behrang Mohit, Nathan Schneider, Rishav Bhowmick, Kemal Oflazer, and Noah A Smith. Recall-oriented learning of named entities in arabic wikipedia. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 162–173, 2012.
17. Hoifung Poon and Pedro Domingos. Unsupervised semantic parsing. In *Proceedings of the 8th Conference on Empirical Methods in Natural Language Processing*, pages 1–10, 2009.
18. Sujan Kumar Saha, Pabitra Mitra, and Sudeshna Sarkar. Word clustering and word selection based feature reduction for maxent based hindi ner. In *Proceedings of the Conference on Human Language Translations*, pages 488–495. Association for Computational Linguistics, 2008.
19. Isaac G Councill, C Lee Giles, and Min-Yen Kan. ParsCit: an open-source CRF reference string parsing package. In *Proceedings of The 6th International Conference on Language Resources and Evaluation*, 2008.

20. Ted Briscoe, John Carroll, and Rebecca Watson. The second release of the rasp system. In *Proceedings of the Joint Conference of the International Committee on Computational Linguistics and the Association for Computational Linguistics*, pages 77–80, 2006.
21. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of Advances in Neural Information Processing Systems 26*, pages 3111–3119, 2013.
22. Kathy McKeown, Hal Daume, Snigdha Chaturvedi, John Paparrizos, Kapil Thadani, Pablo Barrio, Or Biran, Suvarna Bothe, Michael Collins, Kenneth R Fleischmann, et al. Predicting the impact of scientific concepts using full-text features. *Journal of the Association for Information Science and Technology*, 2016.
23. Michael Collins. Head-driven statistical models for natural language parsing. *Computational linguistics*, 29(4):589–637, 2003.
24. Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *Explorations Newsletter from The Association for Computing Machinery's Special Interest Group on Knowledge Discovery and Data Mining*, 11(1):10–18, 2009.
25. Sidney Siegel. *Nonparametric statistics for the behavioral sciences.* McGraw-Hill, 1956.
26. Michael Lesk. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of The 5th Annual International Conference on Systems Documentation*, pages 24–26. Association for Computing Machinery, 1986.
27. Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *In Proceedings of The 7th International Conference on Language Resources and Evaluation*, volume 10, pages 2200–2204, 2010.
28. Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 417–424, 2002.
29. Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, volume 4, pages 412–418, 2004.
30. James R Curran, Stephen Clark, and Johan Bos. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions*, pages 33–36, 2007.
31. Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. In *Proceedings of the International Conference on Machine Learning*, volume 14, pages 1188–1196, 2014.
32. Angelika Kratzer. *Modality. In von Stechow & Wunderlich (eds.), Semantics: An international handbook of contemporary research.* de Gruyter, 1991.
33. Ben Medlock and Ted Briscoe. Weakly supervised learning for hedge classification in scientific literature. In *Proceedings of the Association for Computational Linguistics*, volume 2007, pages 992–999, 2007.
34. Josef Ruppenhofer and Ines Rehbein. Yes we can!? annotating the senses of english modal verbs. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 24–26, 2012.

35. Mengfei Zhou, Anette Frank, Annemarie Friedrich, and Alexis Palmer. Semantically enriched models for modal sense classification. In *Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics (LSDSem)*, page 44, 2015.
36. Ana Marasović and Anette Frank. Multilingual modal sense classification using a convolutional neural network. *In Proceedings of the 1st Workshop on Representation Learning for NLP*, 2016.
37. Janyce Wiebe, Theresa Wilson, and Claire Cardie. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2):165–210, 2005.
38. Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, 2005.
39. Maria Becker, Alexis Palmer, and Anette Frank. Clause types and modality in argumentative microtexts. *In Workshop on Foundations of the Language of Argumentation (in conjunction with COMMA)*, 2016.