*Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pages 6045–6058
Marseille, 20-25 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

# Problem-solving Recognition in Scientific Text

**Kevin Heffernan, Simone Teufel**
University of Cambridge
Dept. of Computer Science and Technology
15 JJ Thomson Ave, Cambridge CB3 0FD
{firstname.lastname}@cst.cam.ac.uk

**Abstract**

As far back as Aristotle, problems and solutions have been recognised as a core pattern of thought, and in particular of the scientific method. In this work, we present the novel task of problem-solving recognition in scientific text. Previous work on problem-solving either is not computational, is not adapted to scientific text, or has been narrow in scope. This work provides a new annotation scheme of problem-solving tailored to the scientific domain. We validate the scheme with an annotation study, and model the task using state-of-the-art baselines such as a Neural Relational Topic Model. The agreement study indicates that our annotation is reliable, and results from modelling show that problem-solving expressions in text can be recognised to a high degree of accuracy.

**Keywords:** Problem-solving, Information Extraction, Machine Learning, Neural Relational Topic Model

## 1. Introduction

In the field of cognitive psychology, problem-solving is formally defined as: *"cognitive processing directed at achieving a goal for which the problem solver does not initially know a solution method"* (Reisberg and Mayer, 2013). Many of us perform it on a daily basis, whether it be deciding on the best route home, what meal to cook for dinner, or even how to structure our day. It is generally regarded as the most important cognitive activity in everyday and professional contexts (Jonassen, 2000).

According to Jordan (1980), this activity carries over from everyday activities, to any text produced. In particular, there is a close connection to scientific writing because the nature of the research process can be viewed as a problem-solving activity (Popper, 1999; Strübing, 2007). Therefore, problem-solving plays a significant role in the understanding of academic texts from the scientific domain, and many descriptions relating to problems and solutions can be found in scientific texts. Consider the following extract, taken from Benotti and Denis (2011).

---

Semantic annotation and rule authoring have long been known as bottlenecks for developing conversational systems for new domains.

In this paper, we present a novel algorithm for generating virtual instructors from automatically annotated human-human corpora.

---

In the above extract, the first sentence details a problem. The authors then introduce an algorithm which is able to use automatically annotated texts and thus solve the problem of expensive manual anntotation.

In this work, we set out to automatically identify such problems and their corresponding solutions in scientific documents. Capturing knowledge of such problem-solving expressions would provide a deep insight into text understanding, but problem-solving is a non-trivial subjective task. For these kinds of tasks, particularly if they are newly defined, there is general consensus that careful human annotation is necessary.

We developed a scheme of problem-solving, tailored to the scientific domain. The particular scheme we use is an adaptation of that by Hoey (1983; 2001). We also developed a finer annotation of problems that gives insight into which aspect of the problem might be seen as the source (or the more immediate manifestation) of the problem. This would result in the earlier problem annotated as follows:

---

Semantic annotation and rule authoring have long been known as bottlenecks for developing conversational systems for new domains.

---

How would a human recognise that there is a problem in the above sentence? Consider the word "bottlenecks" (shown in yellow here), which is a lexical signal for a problematic situation. Also notice that the red string is marking the artefact that would need to be changed in order to solve the problem, whereas the green string expresses some conditions related to the problem. If we found a solution which provides an artefact closely related to the red string, it should have higher probability of solving the problem than some less related potential solution. Therefore, such a subdivision of problem strings should help linking problems with their correct solutions, and our suggested solution will take advantage of this fact. The rest of this paper introduces the new task of problem-solving, our annotation scheme and agreement study, and an automatic baseline method for finding and linking problem and solution descriptions in text.

## 2. Background and related work

Van Dijk (1980) states that all texts have a macrostructure: a semantic characterisation of discourse structures on a global level, representing the entirety of the text. In general scientific discourse, Van Dijk (1980) assigns the macrostructure of INTRODUCTION, PROBLEM, SOLUTION, and CONCLUSION. This is a problem-solving macrostructure and there are many other theorists who agree (Hutchins, 1977; Grimes, 1975).

One of the most well documented problem-solving structures was established by Winter (1968). 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. The crucial refinement here when compared to Van Dijk is the replacement of CONCLUSION with EVALUATION; Winter realised that the SOLUTION needs to be evaluated before being accepted. Hoey (1983) further improved the pattern by using RESPONSE in place of SOLUTION. Hoey's definition of RESPONSE is *any* way to deal with an issue. This has a better semantic fit to scientific texts, where a RESPONSE cannot become a SOLUTION unless it has a positive EVALUATION. The pattern used by Hoey is therefore the variant of the problem-solving pattern upon which we have chosen to base our approach.

Computationally, there are few works that address the identification of problems and solutions directly. Heffernan and Teufel (2018) introduced models for detecting the presence of problems or solutions at the sentence level, and explored various features for this task such as modality, subcategorisation, and word embeddings. They managed to identify problems and solutions with accuracies of 82.3% and 79.7% respectively.

Sasaki et al. (2019) then extended this work by modelling problems and solutions using a transformer decoder (Liu et al., 2018). The transformer decoder problem recogniser achieved a higher accuracy result than Heffernan and Teufel ($\uparrow$ 0.05), but their solution recogniser was numerically lower ($\downarrow$ 0.05).

Problem-solving has also been treated in the framework of discourse theories such as Rhetorical Structure Theory (Mann and Thompson, 1988). Rhetorical Structure Theory (RST) is a theory which aims at describing both the micro- and macro-structures of text by assigning relations between text spans. In this theory, most text spans share a relationship where one text span has a specific role in relation to the other. One such relation in RST is the *solutionhood* relation, which captures both a PROBLEM and its related SOLUTION in text. There have been many attempts at automating the learning of RST relations using discourse parsing (Hernault et al., 2010; Feng and Hirst, 2014; Ji and Eisenstein, 2014; Braud et al., 2017; Mabona et al., 2019), which are all benchmarked on the RST Discourse Treebank dataset (RST-DT) (Carlson et al., 2003).

Beyond these works, most prior research on computational problem-solving has not gone beyond the usage of keyword analysis and some simple contextual examination of the problem-solving pattern. Flowerdew (2008) presents a corpus-based analysis of lexiogrammatical patterns for PROBLEM and SOLUTION clauses using articles from professional and student reports. A large part of the study involved looking for keywords which signalled a PROBLEM or SOLUTION statement.

Scott (2001) also looked at signals of PROBLEM and SOLUTION in the Guardian newspaper. One of the goals of the work was to find signals of problem-solving patterns by automatic methods. Scott started with simple signals such as "problem" and "problems" and then used Mutual Information within a 10-word window of these keywords to see if other indicative words appeared. Another aim in that work was to determine if the signals used for problem-solving patterns shape the text on a macro or micro level (i.e., do the problem-solving patterns encompass and shape the whole text or just account for a small sub-part of the text). He discovered that the latter was the case, where the signals used for problem-solving patterns only play a role at a local level of discourse.

Instead of a keyword-based approach, Charles (2011) used discourse markers to examine how problem-solution patterns are signalled in theses from the domains of politics and materials science. In particular, he examined how the combination of "however" and "thus" can be used in conjunction to signal a problem-solution pattern. It was hypothesised that "however" would signal a PROBLEM, which should be followed by "thus" signalling a SOLUTION. Charles indeed found that these two discourse markers signal a problem-solving pattern in 80% of cases in the corpus.

In comparison to approaches which directly address problems and solutions, the task of problem-solving shares a close connection with other popular NLP tasks. In Argument Mining, two key tasks involve the identification of arguments such as premise and conclusion, and determining the relations between these arguments (Lawrence and Reed, 2020; Cabrio and Villata, 2018). This has certain parallels to finding arguments of problems and solutions, and determining whether the relationship between the solution and problem is that of solved, or not solved.

In the task of Cause-Effect Analysis, the text is analysed to determine which sentences contain cause-effect patterns (Mueller and Huettemann, 2018; Mueller and Abdullaev, 2019), and when describing a problematic situation is it commonplace to describe how the problem manifested (the effect), and the reason the problem arose (the cause).

Sentiment mining is another task with connections to problem-solving, which is concerned with identifying the polarity of a piece of text, with wide applica-

tions across both academia (Li et al., 2017; Liu, 2017) and industry (Asur and Huberman, 2010; Valdivia et al., 2017). In particular, words with negative and positive polarity can be seen as characteristics of problem and solution descriptions respectively. For example, "poor" and "excellent" are words which have traditionally been good indicators of polarity status in many studies (Turney, 2002; Mullen and Collier, 2004).

## 3. Annotating problem-solving in science

In this section, we define our annotation scheme for problem-solving in scientific text. It is based on the theoretic problem-solving model introduced by Hoey (1983), but tailored to the scientific domain. Our scheme is composed of the three main elements:

1. Problems.
2. Solutions.
3. Problem-solution links.

We will now define each element in turn, and also provide motivation and examples of how each element is expressed in text.

### 3.1. Definition of problem

In Hoey's model, a problem is simply defined as: "an issue which requires attention". However, in scientific discourse, both the meaning of "problem" and how it is expressed can vary greatly. In our scheme, a problem is defined as:

1. A problematic state in science; or
2. A research question or unexplained phenomena; or
3. An artefact that does not fulfil its stated specification.

where an artefact, as defined here, is *any tool created by the author to solve some problem, and covers descriptions or namings of methods, frameworks, algorithms, or pseudocode*.

The first category of problems above treats problematic states in science. Such utterances can encompass a wide range of expressions describing an event with negative sentiment. Consider the following examples.

(A) We have to take even more factors into account, but it is difficult to maintain such heuristic rules. (W03-1024, S-20)[1]

(B) Polysemy contributes heavily to poor precision. (W00-0102, S-4)

[1]Each example in this work comes from our corpus' train and development set, described in section 4. Examples in running text are identified with an uppercase letter, and end with the paper's ACL Anthology ID and corresponding sentence number containing the example e.g. (W09-0403, S-1).

The examples above describe some problematic state, be it a difficulty in doing something desired (ex. A) or a low value in an evaluation metric that should ideally be high (ex. B).

The second category of problems concerns research questions or unexplained phenomena. These are considered problems as they imply a state of uncertainty from the viewpoint of the author, and therefore fall under a problematic situation. Ex. C is an instance of research question, and ex. D is an instance of an unexplained phenomena.

(C) But do word-based vectors also work well for genre detection? (E99-1019, S-5)

(D) However, a language independent phoneme set has not been explored yet experimentally. (P05-1064, S-21)

In general, lack of knowledge is always viewed as a problem in the scientific endeavour. The last type of problem concerns bad situations of a particular kind where something is missing. This is often expressed as a need, requirement or lack of something.

(E) Second, those recognizers require large bodies of training data. (W06-3325, S-12)

**Subdividing problems.** Although knowledge of a problem by itself provides valuable information, we aimed to provide a deeper level of problem understanding, such as identifying the immediate source of the problem. We provide one such explanatory division of problem descriptions in science.

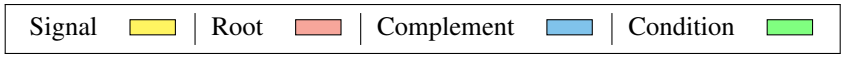
In our scheme, a problem description is subdivided into four elements: SIGNAL, COMPLEMENT, ROOT, and CONDITION. These four elements are defined as follows:

- SIGNAL: a short phrase indicating a problematic situation.
- ROOT: a phrase describing how the problem manifests itself.
- COMPLEMENT: artefact, object, or process affected by the problem; or further description of the problem signal.
- CONDITION: conditions related to the problem.

In order to illustrate how these four elements are realised in running text, Table 1 provides examples of various problem types broken up into SIGNAL, ROOT, COMPLEMENT, and CONDITION. Note that for each problem type, the role played by CONDITION does not change.

| | Problem type | Signal | Root | Complement |
|---|---|---|---|---|
| 1 | **X is lacking/missing from Y** | Signal of the lack | Agent, object, or action experiencing the lack | Object or quality that is lacking or missing |
| 2 | **X has a bad property** | Signal of the bad property | Agent, object, or action possessing the bad property | Additional description of signal |
| 3 | **X cannot obtain objective** | Signal of the inability to attain | Agent which cannot attain objective | Objective which cannot be attained |
| 4 | **X is a bad fit for objective** | Signal of the bad fit | Agent, object, or action which is a bad fit | Additional description of signal |
| 5 | **X performs action badly** | Signal of badly performed action | Agent performing the action | Action performed |
| 6 | **X leads to a problematic situation** | Signal of problematic situation | Agent, object, or root which induces the problematic situation | Object which is made problematic |
| 7 | **Research question** | Signal of problem or uncertainty | The research question | Additional description of signal |

(a) Examples of problem types and the corresponding roles played by SIGNAL, ROOT, and COMPLEMENT.




1 In contrast, formally syntax-based grammars often lack explicit linguistic constraints . (W08-0403, S-14)

2 Out-of-vocabulary (OOV) terms are particularly problematic . (W10-4008, S-12)

3 However, the traditional N-gram language model can not capture long-distance word relations . (P12-1023, S-32)

4 As a result, the common word strategy may not be appropriate for the problem we study here . (P98-1098, S-15)

5 These algorithms tend to approximate the state space excessively . (W08-0120, S-9)

6 Using entity names severely pollutes the embeddings of words . (D15-1031, S-14)

7 To give a specific example, it is not clear yet when L1 structures lead to interference and when they do not . (W13-2606, S-11)

(b) Annotated instances for each example problem type above.

Table 1: Problem types and examples of usage.

These four elements are structured together into a PROBLEM STATEMENT. We define a problem statement to be present if there is a SIGNAL; additionally a problem statement can contain at most one of each of the following: a ROOT, a COMPLEMENT, and a CONDITION. We will from now on represent these four elements by colours: yellow for SIGNAL, red for ROOT, blue for COMPLEMENT, and green for CONDITION.

### 3.2. Definition of solution

A SOLUTION, according to Hoey, is comprised of two elements: a RESPONSE and an accompanying positive EVALUATION. However, Hoey's definition of RESPONSE is very general: *some reaction to a problem which aims at overcoming a problematic situation*. To provide more specificity to science, we expand Hoey's definition of RESPONSE to also include: *a description and/or naming of an artefact contributed by an author in response to a problem*. We consider artefacts a RESPONSE element as there must have been a problematic situation which motivated the authors to create such an artefact (e.g. a negative property of a previously published method or a lack of research). In our scheme, a SOLUTION using this expanded definition of

RESPONSE is defined as:

1. A description of a reaction to a problem aimed at overcoming a problematic situation which is associated with a positive evaluation [Hoey's definition]; or

2. A description and/or naming of an artefact contributed by an author in response to a problem[2].

Consider the following example of SOLUTION below, highlighted in purple.

(F) In this paper, we propose a Bayesian approach called TopicSpam for deceptive review detection. (P13-2039, S-27)

### 3.3. Definition of problem-solution link

A problem-solution link represents a solution-hood relationship between a problem statement and a solution. In the scheme designed by Hoey (1983), this link is implicitly defined when a RESPONSE to a problem is assigned a positive EVALUATION. However, for computational purposes a more precise definition is needed. For example, it is unclear how to link when a SOLUTION solves more than one PROBLEM or if the RESPONSE is positively evaluated but does not fully solve the PROBLEM as it is described. Therefore, as with the definitions for PROBLEM and SOLUTION, we have had to adapt the definition for the link between problems and solutions. We define a problem-solution link as:

> A binary link between any pair of solution and problem statements, where the solution either entirely or partially solves the problematic situation arising from the problem statement, or provides a workaround.

Consider the following example of a link between problem and solution, where the link is represented as LINK signal text.

(G) **1.** Thus, other ways of deciding when to stop AL are LINK needed. **2.** In this paper, we propose an intrinsic stopping-criterion for committee-based AL of named entity recognizers. (W09-1118, [S-26, S-27])

In the above example, the problem stated is that there is a need for a way to decide when to stop AL (Active Learning). In the following sentence, the authors introduce exactly such a solution which is a stopping-criterion for AL. Since this solves the need, the problem must be linked to the solution.

[2]Note that the requirement of an explicitly stated evaluation is lifted here as an author presenting a published artefact implies it has been evaluated positively to pass peer review.

## 4. Data collection

The basis for our data set was a collection of over 17k academic papers extracted from the ACL Anthology Reference Corpus (Bird et al., 2008).

Given the large search space involved in research papers, we only treat the labelling of problems and solutions within a predefined context window[3]. In order to improve the recall of problems or solutions in such context windows, we filtered out any contexts from papers which did not have a high likelihood of problem and solution present. This was established using existing classifiers capable of recognising the presence of a problem or solution (Heffernan and Teufel, 2018), and filtering out contexts which did not have at least one problem and solution identified. From the resulting contexts, we randomly sampled a data set consisting of 1000 contexts (6,000 sentences) with a 80/10/10 split for training, development, and test sets.

**Human agreement study.** The human agreement study consisted of three annotators, comprising one of the authors and two external annotators. In order to train the external annotators, we created an instruction set which can act as a stand-alone document, i.e., it contains all the information needed to complete the task[4]. Following training, each annotator was given a copy of the development set (previously unseen). The annotators then marked each extract from the development set independently and without discussion.
The results from annotation show good agreement amongst the annotators. The kappa value for problem marking was $\kappa$**=0.87** ($N$=6512; $n$=8; $k$=3)[5], solutions were $\kappa$**=0.99** ($N$=2599; $n$=3; $k$=2), and problem-solution linking was $\kappa$**=0.66** ($N$=279; $n$=3; $k$=2). For both problem and solution marking, such kappas can be considered to represent "good agreement" using Krippendorff's strict scale (Krippendorff, 1980).

Following the annotation experiment, the test set was annotated jointly by the two external annotators. The resulting dataset and full annotation instructions are publicly available[6]. Dataset statistics are shown in Table 2.

## 5. Method

In order to determine the tractability of modelling problem-solving, we implement a series of baselines ranging from heuristics to state-of-the-art deep learning models. We split the task of problem-solving into two

[3]A context window of 6 sentences was chosen based on manual exploration of the training set.

[4]In consideration of ethics regarding the dataset collection process and conditions, we applied for and received ethics approval from our institution.

[5]$N$, $n$, and $k$ correspond to number of items to annotate, number of categories and number of annotators, respectively.

[6]https://github.com/kevinheffernan/problem-solving-in-scientific-text/blob/main/dataset.tar.gz

| | Number annotated |
|---|---|
| Words | 166,409 |
| Sentences | 6,000 |
| Signal | 3,072 |
| Root | 2,156 |
| Complement | 1,653 |
| Condition | 971 |
| Solution | 1,018 |
| Problem-solution pairs | 3,472 |

Table 2: Dataset statistics.

main tasks: problem and solution marking, and problem and solution linking. We present the methods for each below, including hyperparmeter tuning.

**Problem and solution marking.** We formulate the marking of problems and solutions as a sequence labelling task, similar to named entity recognition (NER). We therefore encode problems and solutions using the IOB2 encoding scheme (commonly referred to as BIO) (Tjong Kim Sang and Veenstra, 1999), and compare three different methods which have shown to perform well on NER benchmarks. The first two are deep learning models: the Biaffine model by Yu et al. (2020), which is the current state-of-the-art, and a BiLSTM-CRF model (Reimers and Gurevych, 2017a). We used an existing code repository for both the Biaffine[7] and BiLSTM-CRF[8] models. For a simple baseline, we chose a 1st order HMM[9] as a bi-gram HMM has shown to be effective on NER-related tasks (Zhao, 2004).

We use a joint learning approach to problem and solution marking. This is akin to multi-task learning (Ruder, 2017), where it has been observed that learning multiple tasks at the same time can improve performance i.e., the learning of one task can help the performance of another task (Ruder et al., 2019; Sanh et al., 2019).

**Problem and solution linking.** We now move to the second task of linking problems and solutions. Problem and solution linking is formalised as a binary classification task, where a problem and solution pair are given, and a decision is made by the system as to whether or not they are linked. For this binary classification task, we compare two deep learning models, two classical machine learning models, and a most frequent category baseline.

For the classical machine learning models, we opted for Naïve Bayes and a Support Vector Machine. For both classifiers, we used an implementation from the WEKA machine learning library (Hall et al., 2009).

Moving to the neural models, the first is a Neural Relational Topic Model (NRTM) proposed by Bai et al. (2018). This model jointly trains a neural topic model, combined with a feedforward multilayer network (MLP). The neural topic model is implemented as a variational autoencoder (VAE) (Kingma and Welling, 2014) which learns a topic representation for each document. Therefore, this model should in theory be able to match problems and solutions which have a similar topic distribution. As the NRTM is essentially a topic model jointly trained with a multi-layer perceptron (MLP) on top, it is interesting to determine whether or not the joint training with the topic information is able to provide a performance boost over the standalone MLP. Therefore, our second neural model is a standalone MLP.

To implement the NRTM we used an existing code implementation[10], and for the MLP we used our own implementation in Tensorflow consisting of three hidden layers with ReLU (rectified linear unit) activations. The ReLU was chosen as it has been widely successful across many different applications (Ramachandran et al., 2017).

For each model, we compare two features: bag-of-words and Sentence-BERT (Reimers and Gurevych, 2019). For Sentence-BERT (SBERT), we used a pretrained model[11] which achieved the highest performance on the STS Benchmark[12] (Cer et al., 2017).

**Hyperparameter tuning.** For each model (except Naïve Bayes, the HMM, and most-frequent baselines), we conducted a hyperparameter search using manual tuning on our development set[13]. Full details of all hyperparameters used for each model, along with their respective average runtimes and hardware used can be found in the Appendix.

Reimers and Gurevych (2017b) showed that score distributions for neural models can vary greatly depending on the seed value for a random generator. Therefore, all results from neural models are averaged over 50 different random seeds, and are accompanied by their standard deviation. Additionally, for all reported scores, a correct prediction must match the gold standard exactly (e.g. element boundary and label perfectly matches).

In order to measure significance between models, we used the two-tailed permutation test (Noreen, 1989; Dror et al., 2018). All code used in our experiments is

[7] https://github.com/juntaoy/biaffine-ner

[8] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf

[9] https://www.nltk.org/_modules/nltk/tag/hmm.html

[10] https://github.com/zbchern/Neural-Relational-Topic-Models

[11] https://github.com/UKPLab/sentence-transformers/blob/master/docs/pretrained-models

[12] stsb-roberta-large

[13] Highest f1-measure was the selection criterion.

| Element | Biaffine | BiLSTM-CRF | Baseline |
|---|---|---|---|
| Signal | $0.86 \pm 0.01$ | $0.83 \pm 0.01$ | 0.54 |
| Root | $0.66 \pm 0.04$ | $0.63 \pm 0.03$ | 0.13 |
| Complement | $0.59 \pm 0.06$ | $0.57 \pm 0.03$ | 0.10 |
| Condition | $0.60 \pm 0.07$ | $0.57 \pm 0.03$ | 0.11 |
| Problem statement | $0.46 \pm 0.03$ | $0.46 \pm 0.02$ | 0.07 |
| Solution | $0.89 \pm 0.01$ | $0.85 \pm 0.02$ | 0.12 |
| Micro | $0.77 \pm 0.01$ | $0.73 \pm 0.01$ | 0.29 |
| Macro | $0.72 \pm 0.02$ | $0.69 \pm 0.02$ | 0.20 |

(a) F-measures from jointly learning problems and solutions.

| Element | p-value |
|---|---|
| Signal | **0.002** |
| Root | **0.033** |
| Complement | 0.099 |
| Condition | 0.060 |
| Problem statement | 0.369 |
| Solution | **0.001** |
| Micro | **0.001** |
| Macro | **0.008** |

(b) Two-tailed p-values between Biaffine and BiLSTM-CRF.

Table 3: Problem and solution marking results.

| Feature | NB | SVM | MLP | NRTM |
|---|---|---|---|---|
| Bow | 0.56 | 0.56 | $0.53 \pm 0.03$ | $0.47 \pm 0.09$ |
| SBERT | 0.53 | 0.56 | $0.53 \pm 0.04$ | - |
| Baseline | 0.39 | | | |

(a) Macro f-measures.

| | Baseline | NB | SVM | MLP |
|---|---|---|---|---|
| NB | **0.003** | - | - | - |
| SVM | **0.003** | 1.000 | - | - |
| MLP | **0.001** | **0.010** | **0.005** | - |
| NRTM | **0.020** | **0.013** | **0.012** | **0.041** |

(b) Two-tailed p-values between models using best respective feature.

Table 4: Linking results.

publicly available.[14]

## 6. Results

The results from problem and solution marking and linking are shown in Tables 3 and 4. For problem and solution marking, the Biaffine model outperformed the BiLSTM-CRF on all measures, and significantly beat the BiLSTM-CRF on most. SOLUTION was identified with the highest average f-measure, whilst the individual elements with the lowest average f-measure were COMPLEMENT and CONDITION. The baseline HMM shows reasonable performance on SIGNAL, but is significantly outperformed by both neural models on all measures (significance tables included in the Appendix).

The linking results indicate that Naïve Bayes and the SVM are the best performers on this task, significantly beating all other models. The NRTM achieved the lowest average score but has the highest variance, including f-measures as high as 0.58 in its score distribution. Therefore, the results from topic modelling are not as reliable as those from other models.

The confusion matrix for the Naïve Bayes model using bag-of-words features is shown in Table 5. It indicates that the model can more accurately identify problems which are linked to solutions, rather than those which are not. For example, 67% of linked problem-solution pairs were correctly identified, whilst only 45% of non-linked pairs were correctly labelled. A similar trend was exhibited by the SVM where 44% of non-linked and 65% of linked problem-solution pairs were identified correctly.

| | | Predicted | |
|---|---|---|---|
| | | Not linked | Linked |
| Gold | Not linked | 53 (45%) | 64 (55%) |
| | Linked | 65 (33%) | 131 (67%) |

Table 5: Confusion matrix for Naïve Bayes (bow).

## 7. Conclusion and future work

In this work, we have presented the novel task of problem-solving recognition in text. We also introduced a new annotated dataset for this task, and implemented a series of baseline methods to test how well the task can be automated. Inter-annotator statistics show good agreement, and results from automation indicate that the task can be tractably modelled, with room for improvement. In future work, we plan to extend our current dataset to also include the subdivision of SOLUTION expressions, in the same manner which did for PROBLEM. We are also interested in how well problem-solving can be modelled in other languages. One such study is currently being undertaken, applying our rubric to Japanese texts.

[14] https://github.com/kevinheffernan/problem-solving-in-scientific-text/blob/main/SupMat__Software.tgz

## 8. Bibliographical References


Asur, S. and Huberman, B. (2010). Predicting the future with social media. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, volume 1, pages 492–499.

Bai, H., Chen, Z., Lyu, M. R., King, I., and Xu, Z. (2018). Neural relational topic models for scientific article analysis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 27–36.

Benotti, L. and Denis, A. (2011). Prototyping virtual instructors from human-human corpora. In *Proceedings of the ACL-HLT 2011 System Demonstrations*, pages 62–67, Portland, Oregon, June. Association for Computational Linguistics.

Bird, S., Dale, R., Dorr, B., Gibson, B., Joseph, M., Kan, M.-Y., Lee, D., Powley, B., Radev, D., and Tan, Y. F. (2008). The ACL Anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May. European Language Resources Association (ELRA).

Braud, C., Coavoux, M., and Søgaard, A. (2017). Cross-lingual RST discourse parsing. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 292–304, Valencia, Spain, April. Association for Computational Linguistics.

Cabrio, E. and Villata, S. (2018). Five years of argument mining: a data-driven analysis. In *IJCAI*, volume 18, pages 5427–5433.

Carlson, L., Marcu, D., and Okurowski, M. E., (2003). *Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory*, pages 85–112. Springer Netherlands, Dordrecht.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

Charles, M. (2011). Adverbials of result: Phraseology and functions in the problem–solution pattern. *Journal of English for Academic Purposes*, 10(1):47–60.

Dror, R., Baumer, G., Shlomov, S., and Reichart, R. (2018). The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia, July. Association for Computational Linguistics.

Feng, V. W. and Hirst, G. (2014). A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 511–521, Baltimore, Maryland, June. Association for Computational Linguistics.

Flowerdew, L. (2008). *Corpus-based analyses of the problem-solution pattern: a phraseological approach*, volume 29. John Benjamins Publishing.

Grimes, J. E. (1975). *The thread of discourse*, volume 207. Walter de Gruyter.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.

Heffernan, K. and Teufel, S. (2018). Identifying problems and solutions in scientific text. *Scientometrics*, 116(2):1367–1382.

Hernault, H., Prendinger, H., Ishizuka, M., et al. (2010). Hilda: A discourse parser using support vector machine classification. *Dialogue & Discourse*, 1(3).

Hoey, M. (1983). *On the Surface of Discourse*. George Allen and Unwin.

Hoey, M. (2001). *Textual interaction: An introduction to written discourse analysis*. Psychology Press.

Hutchins, J. (1977). On the structure of scientific texts. *UEA Papers in Linguistics*, 5(3):18–39.

Ji, Y. and Eisenstein, J. (2014). Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13–24, Baltimore, Maryland, June. Association for Computational Linguistics.

Jonassen, D. H. (2000). Toward a design theory of problem solving. *Educational technology research and development*, 48(4):63–85.

Jordan, M. P. (1980). Short texts to explain problem-solution structures-and vice versa. *Instructional Science*, 9(3):221–252.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR*.

Krippendorff, K. (1980). *Content analysis: an introduction to its methodology.* Sage Publications.

Lawrence, J. and Reed, C. (2020). Argument mining: A survey. *Computational Linguistics*, 45(4):765–818.

Li, Y., Pan, Q., Yang, T., Wang, S., Tang, J., and Cambria, E. (2017). Learning word representations for sentiment analysis. *Cognitive Computation*, 9(6):843–851.

Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., and Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Liu, H. (2017). Sentiment analysis of citations using word2vec. *arXiv preprint arXiv:1704.00177*.

Mabona, A., Rimell, L., Clark, S., and Vlachos, A. (2019). Neural generative rhetorical structure parsing. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2284–2295, Hong Kong, China, November. Association for Computational Linguistics.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse*, 8(3):243–281.

Mueller, R. and Abdullaev, S. (2019). Deepcause: Hypothesis extraction from information systems papers with deep learning for theory ontology learning. In *Proceedings of the 52nd Hawaii international conference on system sciences*.

Mueller, R. M. and Huettemann, S. (2018). Extracting causal claims from information systems papers with natural language processing for theory ontology learning. In *Proceedings of the 51st Hawaii international conference on system sciences*.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418.

Noreen, E. W. (1989). Computer intensive methods for hypothesis testing: An introduction. *Wiley, New York*, 19:21.

Popper, K. R. (1999). *All life is problem solving*. Psychology Press.

Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.

Reimers, N. and Gurevych, I. (2017a). Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Reimers, N. and Gurevych, I. (2017b). Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark, September. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Reisberg, D. and Mayer, R. E. (2013). Problem solving. *The Oxford Handbook of Cognitive Psychology*, June.

Ruder, S., Bingel, J., Augenstein, I., and Søgaard, A. (2019). Latent multi-task architecture learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4822–4829.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Sanh, V., Wolf, T., and Ruder, S. (2019). A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956.

Sasaki, H., Yamamoto, S., Agchbayar, A., Enkhbayasgalan, N., and Sakata, I. (2019). Inter-domain linking of problems in science and technology through a bibliometric approach. In *2019 Portland International Conference on Management of Engineering and Technology (PICMET)*, pages 1–9. IEEE.

Scott, M. (2001). Mapping key words to problem and solution. *Patterns of Text: in Honour of Michael Hoey. Benjamins, Amsterdam*, pages 109–127.

Strübing, J. (2007). Research as pragmatic problem-solving: The pragmatist roots of empirically-grounded theorizing. *The Sage handbook of grounded theory*, pages 580–602.

Tjong Kim Sang, E. F. and Veenstra, J. (1999). Representing text chunks. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 173–179, Bergen, Norway, June. Association for Computational Linguistics.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Valdivia, A., Luzón, M. V., and Herrera, F. (2017). Sentiment analysis in tripadvisor. *IEEE Intelligent Systems*, 32(4):72–77.

Van Dijk, T. A. (1980). *Text and context explorations in the semantics and pragmatics of discourse*. Longman.

Winter, E. O. (1968). Some aspects of cohesion. *Sentence and Clause in Scientific English*.

Yu, J., Bohnet, B., and Poesio, M. (2020). Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online, July. Association for Computational Linguistics.

Zhao, S. (2004). Named entity recognition in biomedical texts using an HMM model. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 87–90, Geneva, Switzerland, August 28th and 29th. COLING.

## A. Hardware used for all experiments

1. Processor: Intel Core i7-6820HQ CPU @ 2.70GHz $\times$ 8.
2. RAM: 32 GB.
3. GPU: AMD Verde.

## B. Average runtimes

Runtimes are displayed in seconds for each model averaged over 3 runs.

| Model | Avg. runtime |
|---|---|
| Biaffine | 720.06 |
| BiLSTM-CRF | 412.00 |
| HMM | 3.90 |
| MLP | 14.33 |
| NRTM | 137.33 |
| SVM | 1082.50 |
| NB | 9.02 |

## C. Hyperparameter settings

Hyperparameter settings for each model used, and bounds searched during hyperparameter tuning.

### C.1. BiLSTM

| | |
|---|---|
| BiLSTM units | 100 |
| BiLSTM layers | 2 |
| BiLSTM dropout | 0.25 |
| max gradient norm | 1.0 |
| char CNN filter widths | [3, 4, 5] |
| filter size | 30 |
| char embedding size | 30 |
| Komninos embedding size | 300 |
| embedding dropout | 0.5 |
| optimiser | Adam |
| learning rate | 0.001 |

Table 6: Parameters for BiLSTM model.

### C.2. NRTM

| | |
|---|---|
| Num topics | 30 [bounds: 10-50] |
| Topic embedding size | 30 [bounds: 10-50] |
| Topic model layer units | [500, 200, 100] |
| MLP layer units | [1000, 250, 50, 8, 1] |
| SBERT embedding size | 1024 |
| optimiser | Adam |
| learning rate | 0.001 |

Table 7: Parameters for NRTM model.

### C.3. Biaffine

| | |
|---|---|
| FFNN hidden units | 150 |
| FFNN depth | 2 |
| FFNN dropout | 0.2 |
| BiLSTM units | 200 |
| BiLSTM layers | 3 |
| BiLSTM dropout | 0.4 |
| max gradient norm | 5.0 |
| char CNN filter widths | [3, 4, 5] |
| filter size | 50 |
| char embedding size | 8 |
| FastText embedding size | 300 |
| BERT embedding size | 1024 |
| embedding dropout | 0.5 |
| optimiser | Adam |
| learning rate | 0.001 |

Table 8: Parameters for Biaffine model.

### C.4. MLP

| | |
|---|---|
| MLP layer units | [500, 250, 50] |
| SBERT embedding size | 1024 |
| optimiser | Nadam |
| learning rate | 0.01 |

Table 9: Parameters for MLP model.

### C.5. SVM

| | |
|---|---|
| Kernel function | PolyKernel |
| C | 1 [bounds: 1-4] |

Table 10: Parameters for SVM.

## D. Human annotator: additional information

Annotators were myself and two other adults recruited from close friends/family. Participants were offered to and did the work voluntarily. Care was taken to ensure their working conditions were comfortable and annotators were allowed to work at their own pace/schedule. Annotators were told the purpose of the annotation study, and how their data would be used.

### D.1. Rubric

For each problem-solving context window given to you, please perform the following:

1. Read the last sentence and determine if there is a solution(s).
2. For each solution found, mark it.
3. For each sentence in the context window (including the last sentence) please perform the actions below.
   (a) Ask yourself, "Somewhere in this sentence, is the author trying to convey a problem?"

   If not, move onto the next sentence.
   (b) Ask yourself, "How many problem statements is the author trying to describe in this sentence?"

   Be on the lookout for a second or third problem statement. Words such as "due to","as" and "because" can sometimes signal another problem statement.
   (c) For each problem statement found, ask yourself the following questions and annotate in that order:
      i. "Can you find a problem signal?" If you cannot find a problem signal, then according to our definition, there is no problem statement so move on to the next problem statement found. If a signal is found, continue with the questions below.
      ii. "Is there a complement?". If so, mark it.
      iii. "Is there a condition?". If so, mark it.
      iv. "Is there a root?". If so, mark it.
   (d) If there was a solution(s) found in the last sentence, then, for each problem statement found, determine if there exists a problem-solution relationship between each problem statement and each solution marked in the last sentence. If there exists such a relationship, mark it.

## E. Additional significance results

### E.1. Two-tailed p-values between HMM and Biaffine

| | |
|---|---|
| Signal | 0.001 |
| Root | 0.001 |
| Complement | 0.001 |
| Condition | 0.002 |
| Problem statement | 0.001 |
| Solution | 0.001 |
| Micro | 0.001 |
| Macro | 0.001 |

Table 11: Two-tailed p-values between HMM and Biaffine.

### E.2. Two-tailed p-values between HMM and BiLSTM-CRF

| | |
|---|---|
| Signal | 0.001 |
| Root | 0.001 |
| Complement | 0.001 |
| Condition | 0.002 |
| Problem statement | 0.002 |
| Solution | 0.002 |
| Micro | 0.003 |
| Macro | 0.001 |

Table 12: Two-tailed p-values between HMM and BiLSTM-CRF.