Identification of Tasks, Datasets, Evaluation Metrics, and Numeric Scores for Scientific Leaderboards Construction

Yufang Hou, Charles Jochim, Martin Gleize, Francesca Bonin and Debasis Ganguly
IBM Research – Ireland
Dublin, Ireland
{yhou|charlesj|mgleize|fbonin|debasga1}@ie.ibm.com

Abstract

While the fast-paced inception of novel tasks and new datasets helps foster active research in a community towards interesting directions, keeping track of the abundance of research activity in different areas on different datasets is likely to become increasingly difficult. The community could greatly benefit from an automatic system able to summarize scientific results, e.g., in the form of a *leaderboard*. In this paper we build two datasets and develop a framework (*TDMS-IE*) aimed at automatically extracting *task, dataset, metric* and *score* from NLP papers, towards the automatic construction of leaderboards. Experiments show that our model outperforms several baselines by a large margin. Our model is a first step towards automatic leaderboard construction, e.g., in the NLP domain.

1 Introduction

Recent years have witnessed a significant increase in the number of laboratory-based evaluation benchmarks in many of scientific disciplines, e.g., in the year 2018 alone, 140,616 papers were submitted to the pre-print repository arXiv[1] and among them, 3,710 papers are under the *Computer Science – Computation and Language* category. This massive increase in evaluation benchmarks (e.g., in the form of shared tasks) is particularly true for an empirical field such as NLP, which strongly encourages the research community to develop a set of publicly available benchmark tasks, datasets and tools so as to reinforce reproducible experiments.

Researchers have realized the importance of conducting meta-analysis of a number of comparable publications, i.e., the ones which use similar, if not identical, experimental settings, from shared tasks and proceedings, as shown by special issues dedicated to analysis of reproducibility in experiments (Ferro et al., 2018), or by detailed comparative analysis of experimental results reported on the same dataset in published papers (Armstrong et al., 2009).

A useful output of this meta-analysis is often a summary of the results of a comparable set of experiments (in terms of the tasks they are applied on, the datasets on which they are tested and the metrics used for evaluation) in a tabular form, commonly referred to as a *leaderboard*. Such a meta-analysis summary in the form of a leaderboard is potentially useful to researchers for the purpose of (1) choosing the appropriate existing literature for fair comparisons against a newly proposed method; and (2) selecting strong baselines, which the new method should be compared against.

Although recently there has been some effort to manually keep an account of progress on various research fields in the form of leaderboards, either by individual researchers[2] or in a moderated crowd-sourced environment by organizations[3], it is likely to become increasingly difficult and time-consuming over the passage of time.

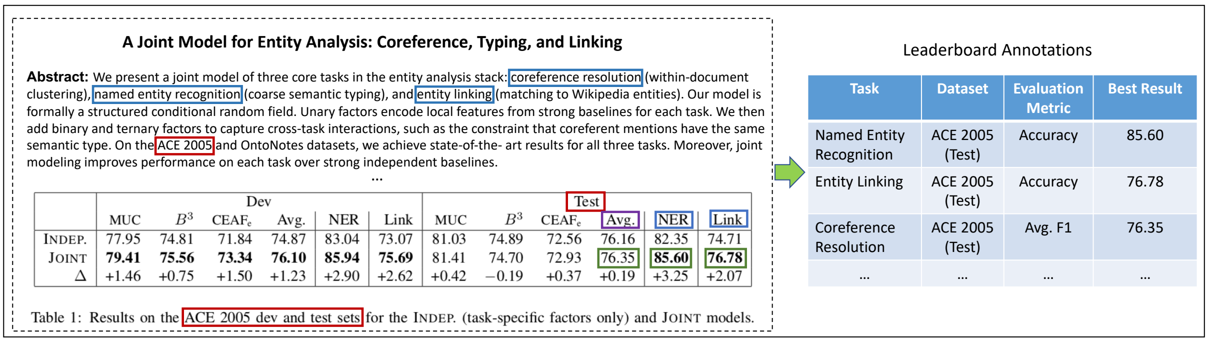
In this paper, we develop a model to automatically identify *tasks*, *datasets*, *evaluation metrics*, and to extract the corresponding *best numeric scores* from experimental scientific papers. An illustrative example is shown in Figure 1: given the sample paper shown on the left, which carries out research work on three different tasks (i.e., *coreference resolution*, *named entity recognition*, and *entity linking*), the system is supposed to extract the corresponding *Task-Dataset-Metric-Score* tuples as shown on the right part in Figure 1. It is noteworthy that we aim to identify a set of pre-

[1]https://arxiv.org/

[2]https://github.com/sebastianruder/NLP-progress

[3]https://paperswithcode.com

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5203–5213
Florence, Italy, July 28 - August 2, 2019.



A Joint Model for Entity Analysis: Coreference, Typing, and Linking

Abstract: We present a joint model of three core tasks in the entity analysis stack: coreference resolution (within-document clustering), named entity recognition (coarse semantic typing), and entity linking (matching to Wikipedia entities). Our model is formally a structured conditional random field. Unary factors encode local features from strong baselines for each task. We then add binary and ternary factors to capture cross-task interactions, such as the constraint that coreferent mentions have the same semantic type. On the ACE 2005 and OntoNotes datasets, we achieve state-of-the- art results for all three tasks. Moreover, joint modeling improves performance on each task over strong independent baselines.

...

	Dev						Test					
	MUC	B^3	CEAF$_e$	Avg.	NER	Link	MUC	B^3	CEAF$_e$	Avg.	NER	Link
INDEP.	77.95	74.81	71.84	74.87	83.04	73.07	81.03	74.89	72.56	76.16	82.35	74.71
JOINT	**79.41**	**75.56**	**73.34**	**76.10**	**85.94**	**75.69**	81.41	74.70	72.93	76.35	**85.60**	**76.78**
Δ	+1.46	+0.75	+1.50	+1.23	+2.90	+2.62	+0.42	−0.19	+0.37	+0.19	+3.25	+2.07

Table 1: Results on the ACE 2005 dev and test sets for the INDEP. (task-specific factors only) and JOINT models.

Leaderboard Annotations

Task	Dataset	Evaluation Metric	Best Result
Named Entity Recognition	ACE 2005 (Test)	Accuracy	85.60
Entity Linking	ACE 2005 (Test)	Accuracy	76.78
Coreference Resolution	ACE 2005 (Test)	Avg. F1	76.35
...	...	...	...

Figure 1: An illustrative example of leaderboard construction from a sample article. The cue words related to the annotated tasks, datasets, evaluation metrics and the corresponding best scores are shown in blue, red, purple and green, respectively. Note that sometimes the cue words appearing in the article are different from the document-level annotations, e.g., Avg. – Avg. F1, NER – Named Entity Recognition.

defined *Task-Dataset-Metric* (*TDM*) triples from a taxonomy for a paper, and the corresponding cue words appearing in the paper could have a different surface form, e.g., *Named Entity Recognition (taxonomy) – Name Tagging (paper)*.

Different from most previous work on information extraction from scientific literature which concentrates mainly on the abstract section or individual paragraphs (Augenstein et al., 2017; Gábor et al., 2018; Luan et al., 2018), our task needs to analyze the entire paper. More importantly, our main goal is to *tag* papers using TDM triples from a taxonomy and to use these triples to organize papers. We adopt an approach similar to that used for some natural language inference (NLI) tasks (Bowman et al., 2015; Poliak et al., 2018). Specifically, given a scientific paper in PDF format, our system first extracts the key contents from the abstract and experimental sections, as well as from the tables. Then, we identify a set of *Task-Dataset-Metric* (*TDM*) triples or *Dataset-Metric* (*DM*) pairs per paper. Our approach predicts if the textual *context* matches the *TDM*/*DM* label *hypothesis*, forcing the model to learn the similarity patterns between the text and various *TDM* triples. For instance, the model will capture the similarities between *ROUGE-2* and "Rg-2". We further demonstrate that our framework is able to generalize to the new (unobserved) TDM triples at test time in a zero-shot TDM triple identification setup.

To evaluate our approach, we create a dataset *NLP-TDMS* which contains around 800 leaderboard annotations for more than 300 papers. Experiments show that our model outperforms several baselines by a large margin for extracting *TDM* triples. We further carry out experiments on a much larger dataset *ARC-PDN* and demonstrate that our system can support the construction of various leaderboards from a large number of scientific papers in the NLP domain.

To the best of our knowledge, our work is the first attempt towards the creation of NLP Leaderboards in an automatic fashion. We pre-process both datasets (papers in PDF format) using GROBID (Lopez, 2009) and an in-house PDF table extractor. The processed datasets and code are publicly available at: `https://github.com/IBM/science-result-extractor`.

2 Related Work

A number of studies have recently explored methods for extracting information from scientific papers. Initial interest was shown in the analysis of citations (Athar and Teufel, 2012a,b; Jurgens et al., 2018) and analysis of the topic trends in the scientific communities (Vogel and Jurafsky, 2012). Gupta and Manning (2011); Gábor et al. (2016) propose unsupervised methods for the extraction of entities such as papers' focus and methodology; similarly, in (Tsai et al., 2013), an unsupervised bootstrapping method is used to identify and cluster the main concepts of a paper. But only in 2017, Augenstein et al. (2017) formalized a new task (SemEval 2017 Task 10) for the identification of three types of entities (called keyphrases, i.e., *Tasks, Methods*, and *Materials*) and two relation types (*hyponym-of* and *synonym-of*) in a corpus of 500 paragraphs from articles in the domains of *Computer Science*, *Material Sciences* and *Physics*. Gábor et al. (2018) also presented the task of IE from scientific papers (Se-

	Macro P	Macro R	Macro F_1
Table caption	79.2	87.0	82.6
Numeric value + IsBolded + Table caption	71.1	77.7	74.0
Numeric value + Row label+ Table caption	55.5	71.4	61.4
Numeric value + Column label + Table caption	49.8	67.2	55.4
Numeric value + IsBolded + Row label + Column label + Table caption	36.6	60.9	43.0

Table 1: Table extraction results of our table parser on 50 tables from 10 NLP papers in PDF format.

mEval 2018 Task 7) with a dataset of 350 annotated abstracts. Ammar et al. (2017, 2018); Luan et al. (2017); Augenstein and Søgaard (2017) exploit these datasets to test neural models for IE on scientific literature. Luan et al. (2018) extend those datasets by adding more relation types and cross-sentence relations using coreference links. The authors also develop a framework called Scientific Information Extractor for the extraction of six types of scientific entities (*Task, Method, Metric, Material, Other-ScientificTerm* and *Generic*) and seven relation types (*Compare, Part-of, Conjunction, Evaluate-for, Feature-of, Used-for*, and *Hyponym-of*). They reach 64.2 F_1 on entity recognition and 39.2 F_1 on relation extraction. Differently from (Luan et al., 2018), (1) we concentrate on the identification of entities from a taxonomy that are necessary for the reconstruction of leaderboards (i.e., *task, dataset, metric*); (2) we analyse the entire paper, not only the abstract (the reason being that the *score* information is rarely contained in the abstract).

Our method for TDMS identification resembles some approaches used for textual entailment (Dagan et al., 2006) or natural language inference (NLI) (Bowman et al., 2015). We follow the example of White et al. (2017) and Poliak et al. (2018) who reframe different NLP tasks, including extraction tasks, as NLI problems. Eichler et al. (2017) and Obamuyide and Vlachos (2018) have both used NLI approaches for relation extraction. Our work differs in the information extracted and consequently in what *context* and *hypothesis* information we model. Currently, one of the best performing NLI models (e.g., on the SNLI dataset) for three way classification is (Liu et al., 2019). The authors apply deep neural networks and make use of BERT (Devlin et al., 2019), a novel language representation model. They reach an accuracy of 91.1%. Kim et al. (2019) exploit densely-connected co-attentive recurrent neural network, and reach 90% accuracy. In our scenario, we generate pseudo premises and hypotheses, then apply the standard transformer encoder (Ashish et al., 2017; Devlin et al., 2019) to train two NLI models.

3 Dataset Construction

We create two datasets for testing our approach for *task, dataset, metric, and score* (TDMS) identification. Both datasets are taken from a collection of NLP papers in PDF format and both require similar pre-processing. First, we parse the PDFs using GROBID (Lopez, 2009) to extract the title, abstract, and for each section, the section title and its corresponding content. Then we apply an improved table parser we developed, built on GROBID's output, to extract all tables containing numeric cells from the paper. Each extracted table contains the table caption and a list of numeric cells. For each numeric cell, we detect whether it has a bold typeface, and associate it to its corresponding row and column headers. For instance, for the sample paper shown in Figure 1, after processing the table shown, we extract the bolded number "85.60" and find its corresponding column headers "{Test, NER}".

We evaluated our table parser on a set of 10 papers from different venues (e.g., *EMNLP, Computational Linguistics journal*). In total, these papers contain 50 tables with 1,063 numeric content cells. Table 1 shows the results for extracting different table elements. Our table parser achieves a macro F_1 score of 82.6 for identifying table captions, and 74.0 macro F_1 for extracting tuples of $<$*Numeric value, Bolded Info, Table caption*$>$. In general, it obtains higher recall than precision in all evaluation dimensions.

In the remainder of this section we describe our two datasets in detail.

3.1 NLP-TDMS

The content of the NLP-progress Github repository[4] provides us with expert annotations of various leaderboards for a few hundred papers in the

[4] https://github.com/sebastianruder/NLP-progress

	Full	Exp
Papers	332	332
Extracted tables	1269	1269
"Unknown" annotations	-	90
Leaderboard annotations	848	606
Distinct leaderboards	168	77
Distinct tasks	35	18
Distinct datasets	99	44
Distinct metrics	72	30

Table 2: Statistics of leaderboard annotations in *NLP-TDMS (Full)* and *NLP-TDMS (Exp)*.

NLP domain. The repository is organized following a "language-domain/task-dataset-leaderboard" structure. After crawling this information together with the corresponding papers (in PDF format), we clean the dataset manually. This includes: (1) normalizing task name, dataset name, and evaluation metrics across leaderboards created by different experts, e.g., using "F1" to represent "F-score" and "Fscore"; (2) for each leaderboard table, only keeping the best result from the same paper[5]; (3) splitting a leaderboard table into several leaderboard tables if its column headers represent datasets instead of evaluation metrics.

The resulting dataset *NLP-TDMS (Full)* contains 332 papers with 848 leaderboard annotations. Each leaderboard annotation is a tuple containing *task, dataset, metric*, and *score* (as shown in Figure 1). In total, we have 168 distinct leaderboards (i.e., <*Task, Dataset, Metric*> triples) and only around half of them (77) are associated with at least five papers. We treat these manually curated *TDM* triples as an NLP knowledge taxonomy and we aim to explore how well we can associate a paper to the corresponding *TDM* triples.

We further create *NLP-TDMS (Exp)* by removing those leaderboards that are associated with fewer than five papers. If all leaderboard annotations of a paper belong to these removed leaderboards, we tag this paper as "Unknown". Table 2 compares statistics of *NLP-TDMS (Full)* and *NLP-TDMS (Exp)*. All experiments in this paper (except experiments in the zero-shot setup in Section 7) are on *NLP-TDMS (Exp)* and going forward we will refer to that only as *NLP-TDMS*.

[5]In this paper, we focus on tagging papers with different leaderboards (i.e., *TDM* triples). For each leaderboard table, an ideal situation would be to extract all results reported in the same paper and associate them to different *methods*, we leave this for future work.

	#Papers	#Extracted tables
ACL	1958	4537
EMNLP	1167	3488
NAACL	730	1559
Total	3855	9584

Table 3: Statistics of papers and extracted tables in *ARC-PDN*.

3.2 ARC-PDN

To test our model in a more realistic scenario, we create a second dataset *ARC-PDN*.[6] We select papers (in PDF format) published in ACL, EMNLP, and NAACL between 2010 to 2015 from the most recent version of the ACL Anthology Reference Corpus (ARC) (Bird et al., 2008). Table 3 shows statistics about papers and extracted tables in this dataset after the PDF parsing described above.

4 Method for TDMS Identification

4.1 Problem Definition

We represent each leaderboard as a <*Task, Dataset, Metric*> triple (*TDM* triple). Given an experimental scientific paper D, we want to identify relevant *TDM* triples from a taxonomy and extract the best numeric score for each predicted *TDM* triple.

However, scientific papers are often long documents and only some parts of the document are useful to predict *TDM* triples and the associated scores. Hence, we define a document representation, called *DocTAET* and a table *score* representation, called *SC* (score context), as follows:

DocTAET. For each scientific paper, its *DocTAET* representation contains the following four parts: *Title, Abstract, ExpSetup*, and *TableInfo*. *Title* and *Abstract* often help in predicting *Task*. *ExpSetup* contains all sentences which are likely to describe the experimental setup, which can help to predict *Dataset* and *Metric*. We use a few heuristics to extract such sentences.[7] Finally, table captions and column headers are important in predicting *Dataset* and *Metric*. We collect them in the

[6]PDN comes from the anthology's directory prefixes for ACL, EMNLP, and NAACL, respectively.

[7]A sentence is included in *ExpSetup* if it: (1) contains any of the following cue words/phrases: {*experiment on, experiment in, evaluation(s), evaluate, evaluated, dataset(s), corpus, corpora*}; and (2) belongs to a section whose title contains any of the following words: {*experiment(s), evaluation, dataset(s)*}.

TableInfo part. Figure 2 (upper right) illustrates the *DocTAET* extraction for a given paper.

SC. For each table in a scientific paper, we focus on boldfaced numeric scores because they are more likely to be the best scores for the corresponding *TDM* triples.[8] For a specific boldfaced numeric score in a table, its context (*SC*) contains its corresponding column headers and the table caption. Figure 2 (lower right) shows the extracted *SC* for the scores *85.60* and *61.71*.

4.2 TDMS-IE System

We develop a system called *TDMS-IE* to associate *TDM* triples to a given experimental scientific paper. Our system also extracts the best numeric *score* for each predicted *TDM* triple. Figure 3 shows the system architecture for *TDMS-IE*.

4.2.1 TDMS-IE Classification Models

To predict correct *TDM* triples and associate the appropriate scores, we adopt a natural language inference approach (NLI) (Poliak et al., 2018) and learn a binary classifier for pairs of document *contexts* and *TDM* label *hypotheses*. Specifically, we split the problem into two tasks: (1) given a document representation *DocTAET*, we would like to predict whether a specific *TDM* triple can be inferred (e.g., give a document we infer <*Summarization, Gigaword, ROUGE-2*>); (2) we predict whether a <*Dataset, Metric*> tuple (*DM*) can be inferred given a score context *SC*.[9] This setup has two advantages: first, it naturally captures the inter-relations between different labels by encoding the three types of labels (i.e., task, dataset, metric) into the same hypothesis. Second, similar to approaches for NLI, it forces the model to focus on learning the similarity patterns between *DocTAET* and various *TDM* triples. For instance, the model will capture the similarities between *ROUGE-2* and "Rg-2".

Recently, a multi-head self-attention encoder (Ashish et al., 2017) has been shown to perform well in various NLP tasks, including NLI (Devlin et al., 2019). We apply the standard transformer encoder (Devlin et al., 2019) to train our models, one for *TDM* triple prediction, and one for score extraction. In the following we describe how we generate training instances for these two models.

DocTAET-TDM model. Illustrated in Figure 3 (upper left), this model predicts whether a *TDM* triple can be inferred from a *DocTAET*. For a set of n *TDM* triples ($\{t_1, t_2, ..., t_n\}$) from a taxonomy, if a paper d_i (*DocTAET*) is annotated with t_1 and t_2, we then generate two positive training instances ($d_i \Rightarrow t_1$ and $d_i \Rightarrow t_2$) and $n-2$ negative training instances ($d_i \nRightarrow t_j, 2 < j \leq n$).

SC-DM model. Illustrated in Figure 3 (lower left), this model predicts whether a score context *SC* indicates a *DM* pair. To form training instances, we start with the list of *DM* pairs ($\{p_1, p_2, ..., p_m\}$) from a taxonomy and a paper d_i, which is annotated with a *TDM* triple t (containing p_1) and a numeric score s. We first try to extract the score contexts (*SC*) for all bolded numeric scores. If d_i's annotated score s is equal to one of the bolded scores s_k (typically there should not be more than one), we generate a positive training instance ($SC_{s_{k=1}} \Rightarrow p_1$). Negative instances can be generated for this context by choosing other DMs not associated with the context, i.e., $m-1$ negative training instances ($SC_{s_{k=1}} \nRightarrow p_j$, $1 < j \leq m$). For example, an SC with "ROUGE for anonymized CNN/Daily Mail" might form a positive instance with DM <*CNN / Daily Mail, ROUGE-L*>, and then a negative instance with DM <*Penn Treebank, LAS*>. Additional negative training instances come from bolded scores s_k which do not match s (e.g., $SC_{s_k} \nRightarrow p_j$, $1 < k$, $1 \leq j \leq m$).

4.2.2 Inference

During the inference stage (see Figure 3 (right)), for a given scientific paper in PDF format, our system first uses the PDF parser and table extractor (described in Section 3) to generate the document representation *DocTAET*. We also extract all boldfaced scores and their contexts from each table. Next, we apply the *DocTAET-TDM* model to predict *TDM* triples among all *TDM* triple candidates for the paper[10]. Then, to extract scores for the predicted *TDM* triples, we apply the *SC-DM* model to every extracted score context (*SC*) and predicted *DM* pair (taken from the predicted *TDM* triples). This step tells us how likely it is that a

[8]We randomly choose 10 papers from *NLP-TDMS (Full)* and compare their *TDMS* tuple annotations with the results reported in the original tables. We found that 78% (18/23) of the annotated tuples contain boldfaced numeric scores.

[9]We look for the relation *SC-DM*, rather then *SC-TDM*, because rarely the task is mentioned in *SC*.

[10]The *TDM* triple candidates could be the valid *TDM* triples from the training set, or a set of *TDM* triples from a taxonomy.

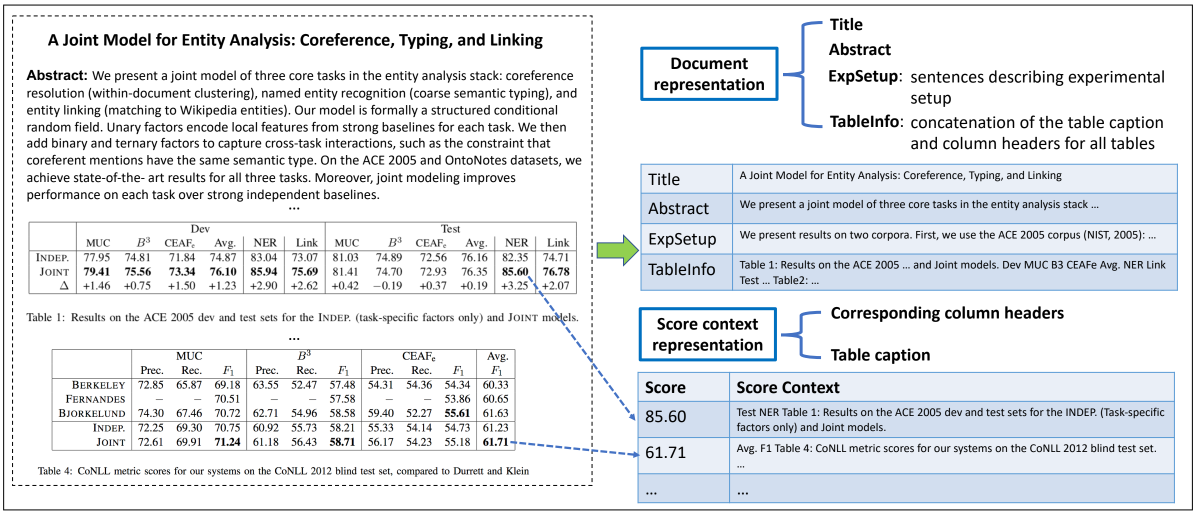


Figure 2: Examples of document representation (*DocTAET*) and score context (*SC*) representation.

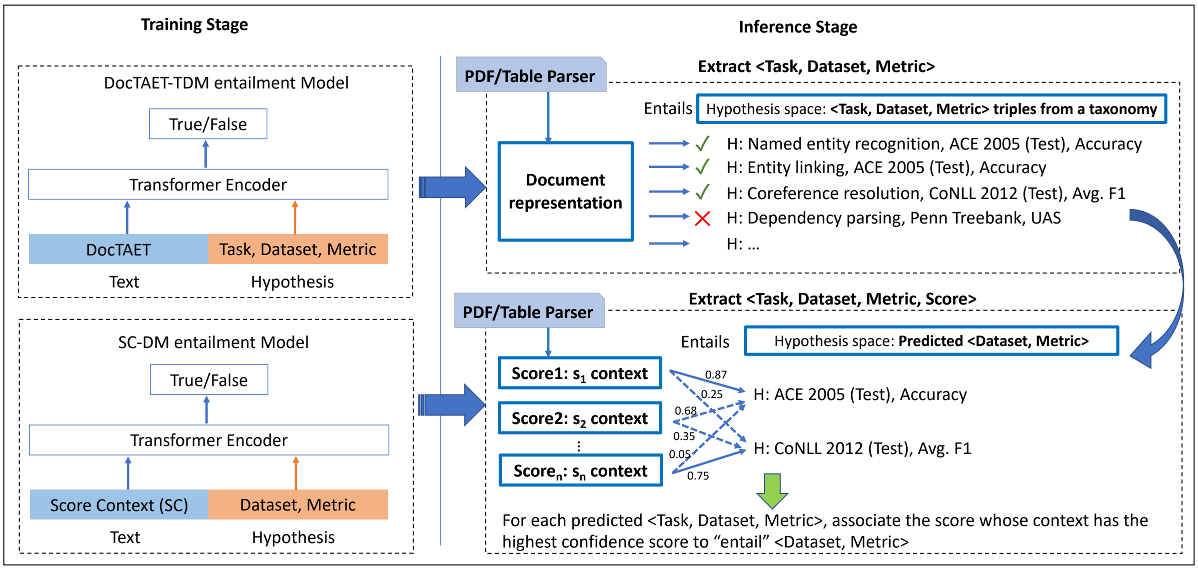


Figure 3: System architecture for *TDMS-IE*.

score context suggests a *DM* pair. Finally, for each predicted *TDM* triple, we select the score whose context has the highest confidence in predicting a link to the constituent *DM* pair.

5 Experimental Setup

5.1 Training/Test Datasets

We split *NLP-TDMS* (described in Section 3) into training and test sets. The partitioning ensures that every *TDM* triple annotated in *NLP-TDMS* appears both in the training and test set, so that a classifier will not have to predict unseen labels (or infer unseen hypotheses). Table 4 shows statistics on these two splits. The 77 leaderboards in this dataset constitute the set of n *TDM* triples we aim to predict (see Section 4.2).

For evaluation, we report macro- and micro-averaged precision, recall, and F_1 score for extracting *TDM* triples and *TDMS* tuples over papers in the test set.

5.2 Implementation Details

Both of our models (*DocTAET-TDM* and *SC-DM*) have 12 transformer blocks, 768 hidden units, and 12 self-attention heads. For *DocTAET-TDM*, we first initialize it using BERT_{BASE}, then fine-tune the model for 3 epochs with the learning rate of $5e-5$. During training and testing, the maximum text length is set to 512 tokens. Note that the document representation *DocTAET* can contain more

	training	test
Papers	170	162
Extracted tables	679	590
“Unknown” annotations	46	44
Leaderboard annotations	325	281
Distinct leaderboards	77	77

Table 4: Statistics of training/test sets in *NLP-TDMS*.

than 1000 tokens for some scientific papers, often due to very long content in *ExpSetup* and *TableInfo*. Therefore, in these cases, we use only the first 150 tokens from *ExpSetup* and *TableInfo* respectively.

We initialize the *SC-DM* model using the trained *DocTAET-TDM* model. We suspect that *DocTAET-TDM* already captures some of the relationship between score contexts and *DM* pairs. After initialization, we continue fine-tuning the model for 3 epochs with the learning rate of $5e-5$. For *SC-DM*, we set a maximum token length of 128 for both training and testing.

5.3 Baselines

In this section, we introduce three baselines against which we can evaluate our method.

StringMatch (SM). Given a paper, for each *TDM* triple, we first check whether the content of the title, abstract, or introduction contains the name of the task. Then we inspect the contexts of all extracted boldfaced scores to check whether: (1) the name of the dataset is mentioned in the table caption and one of the associated column headers matches the metric name; or (2) the metric name is mentioned in the table caption and one of the associated column headers matches the dataset name. If more than one numeric score is identified during the previous step, we choose the highest or lowest value according to the property of the metric (e.g., *accuracy* should be high, while *perplexity* should be low).

Finally, if all of the above conditions are satisfied for a given paper, we predict the *TDM* triple along with the chosen score. Otherwise, we tag the paper as “Unknown”.

Multi-label classification (MLC). For a machine learning baseline, we treat this task as a multi-class, multi-label classification problem where we would like to predict the *TDM* label for a given paper (as opposed to predicting whether we can infer a given *TDM* label based on the paper). The class labels are *TDM* triples and each paper can have multiple *TDM* labels as they may report results from different tasks, datasets, and with different metrics. For this classification we ignore instances with the ‘Unknown’ label in training because this does not form a coherent class (and would otherwise dominate the other classes). Then, for each paper, we extract bag-of-word features with tf-idf weights from the DocTAET representation described in Section 4. We train a multinomial logistic regression classifier implemented in scikit-learn (Pedregosa et al., 2011) using SAGA optimization (Defazio et al., 2014). In this multi-label setting, the classifier can return an empty set of labels. When this is the case we take the most likely *TDM* label as the prediction.

After predicting *TDM* labels we need a separate baseline classifier to compare to the SC-DM model. Similar to the SC-DM model, the MLC should predict the best score based on the *SC*. For training this classifier we form instances from triples of paper, score, and *SC* (as described in Section 4), with a binary label for whether or not this score is the actual leaderboard score from the paper. This version of the training set for classification has 1647 instances, but is quite skewed with only 67 *true* labels. This skew is not as problematic because for this baseline we are not classifying whether or not the SC matches the leaderboard score, but instead we simply pick the most likely SC for a given paper.[11] The scores chosen (in this case one per paper) are combined with the *TDM* predictions above to form the final *TDMS* predictions reported in Section 6.1.

EntityLinking (EL) for *TDM* triples prediction. We apply the state-of-the-art IE system on scientific literature (Luan et al., 2018) to extract *task*, *material* and *metric* mentions from *DocTAET*. We then generate possible *TDM* triples by combining these three types of mentions (note that many combinations could be invalid *TDM* triples). Finally we link these candidates to the valid *TDM* triples in a taxonomy[12] based on Jaccard similarity. Specifically, we predict a *TDM* triple for a paper if the similarity score between the triple and a candidate is greater than α (α is estimated in the

[11]Papers in the test set have an average of 47.3 scores to choose between.

[12]In this experiment, the taxonomy consists of 77 *TDM* triples reported in Table 4.

	Macro P	Macro R	Macro F_1	Micro P	Micro R	Micro F_1
(a) Task + Dataset + Metric Extraction						
SM	31.8	30.6	31.0	36.0	19.6	25.4
MLC	42.0	23.1	27.8	42.0	20.9	27.9
EL	18.1	31.8	20.5	24.3	36.3	29.1
TDMS-IE	**62.5**	**75.2**	**65.3**	**60.8**	**76.8**	**67.8**
(b) Task + Dataset + Metric Extraction (excluding papers with "Unknown" annotation)						
SM	8.1	6.4	6.9	16.8	7.8	10.6
MLC	**56.8**	30.9	37.3	56.8	23.8	33.6
EL	24.9	43.6	28.1	29.4	42.0	34.6
TDMS-IE	54.1	**65.9**	**56.6**	**60.2**	**73.1**	**66.0**
(c) Task + Dataset + Metric + Score Extraction (excluding papers with "Unknown" annotation)						
SM	1.3	1.0	1.1	3.8	1.8	2.4
MLC	6.8	6.1	6.2	6.8	2.9	4.0
TDMS-IE	9.3	11.8	9.9	10.8	13.1	11.8

Table 5: Leaderboard extraction results of *TDMS-IE* and several baselines on the *NLP-TDMS* test dataset.

training set). If none of *TDM* triples was identified, we tag the paper as "Unknown".

6 Experimental Results

6.1 Extraction Results on NLP-TDMS

We evaluate our *TDMS-IE* on the test dataset of *NLP-TDMS*. Table 5 shows the results of our model compared to baselines in different evaluation settings: *TDM* extraction (Table 5a), *TDM* extraction excluding papers with "Unknown" annotation (Table 5b), and *TDMS* extraction excluding papers with "Unknown" annotation (Table 5c).

TDMS-IE outperforms baselines by a large margin in all evaluation metrics for the first two evaluation scenarios, where the task is to extract triples <*Task, Dataset, Metric*>. On testing papers with at least one *TDM* triple annotation, it achieves a macro F_1 score of 56.6 and a micro F_1 score of 66.0 for predicting *TDM* triples, versus the 37.3 macro F_1, and 33.6 micro F_1 of the multi-label classification approach.

However, when we add the *score* extraction (TDMS), even if *TDMS-IE* outperforms the baselines, the overall performances are still unsatisfactory, underlining the challenging nature of the task. A qualitative analysis showed that many of the errors were triggered by the noise from the table parser, e.g., failing to identify bolded numeric scores or column headers (see Table 1). Sometimes a few papers bold the numeric scores for methods from the previous work when comparing to the state-of-the-art results, and our model wrongly predicts these bolded scores for the targeting *TDM* triples.

6.2 Ablations

To understand the effect of *ExpSetup* and *TableInfo* in document representation *DocTAET* for predicting *TDM* triples, we carry out an ablation experiment. We train and test our system with *DocTAET* containing only *Title+Abstract*, *Title+Abstract+ExpSetup*, and *Title+Abstract+TableInfo* respectively. Table 6 reports the results of different configurations for *DocTAET*. We observe that both *ExpSetup* and *TableInfo* are helpful for predicting *TDM* triples. It also seems that descriptions from table captions and headers (*TableInfo*) are more informative than descriptions of experiments (*ExpSetup*).

6.3 Results on ARC-PDN

To test whether our system can support to construct various leaderboards from a large number of NLP papers, we apply our model trained on the *NLP-TDMS* training set to *ARC-PDN*. We exclude five papers which also appear in the training set and predict *TDMS* tuples for each paper.

The set of 77 candidate *TDM* triples comes from the training data, and many of these contain datasets that appear only after 2015. Consequently, fewer papers are tagged with these triples. Therefore, for evaluation we manually choose ten *TDM* triples among all *TDM* triples with at least ten associated papers. These ten *TDM* triples cover various research areas in NLP and contain datasets appearing before 2015. For each chosen *TDM* triple, we rank predicted papers according to the confidence score from the *DocTAET-TDM* model and manually evaluate the top ten results.

Table 7 reports P@1, P@3, P@5, and P@10 for each leaderboard (i.e., *TDM* triple). The macro

Document Representation	Macro P	Macro R	Macro F_1	Micro P	Micro R	Micro F_1
Title+Abstract	11.3	11.3	10.7	47.9	14.2	21.9
Title+Abstract + ExpSetup	20.8	20.1	19.4	50.0	23.7	32.2
Title+Abstract + TableInfo	29.6	29.1	28.1	68.6	40.3	50.8
Title+Abstract + ExpSetup + TableInfo	**62.5**	**75.2**	**65.3**	**60.8**	**76.8**	**67.8**

Table 6: Ablation experiments results of *TDMS-IE* for *Task* + *Dataset* + *Metric* prediction.

Task:Dataset:Metric	P@1	P@3	P@5	P@10	#Correct Score	#Wrong Task
Dependency parsing:Penn Treebank:UAS	1.0	1.0	0.8	0.9	2	0
Summarization:DUC 2004 Task 1:ROUGE-2	0.0	0.67	0.8	0.7	0	0
Word sense disambiguation:Senseval 2:F1	0.0	0.0	0.1	0.1	0	0
Word sense disambiguation:SemEval 2007:F1	1.0	1.0	0.8	0.7	1	0
Word segmentation:Chinese Treebank 6:F1	1.0	0.67	0.4	0.2	0	2
Word Segmentation:MSRA:F1	1.0	0.67	0.6	0.7	2	3
Sentiment analysis:SST-2:Accuracy	1.0	0.67	0.6	0.3	0	3
AMR parsing:LDC2014T12:F1 on All	0.0	0.67	0.4	0.2	0	5
CCG supertagging:CCGBank:Accuracy	1.0	1.0	1.0	0.8	0	1
Machine translation:WMT 2014 EN-FR:BLEU	1.0	0.33	0.2	0.1	0	0
Macro-average	0.70	0.67	0.57	0.46	-	-

Table 7: Results of *TDMS-IE* for ten leaderboards on *ARC-PDN*.

average P@1 and P@3 are 0.70 and 0.67, respectively, which is encouraging. Overall, 86% of papers are related to the target task *T*. We found that most false positives are due to the fact that these papers conduct research on the target task *T*, but report results on a different dataset or use the target dataset *D* as a resource to extract features. For instance, most predicted papers for the leaderboard *<Machine translation, WMT 2014 EN-FR, BLEU>* are papers about *Machine translation* but these papers report results on the dataset *WMT 2012 EN-FR* or *WMT 2014 EN-DE*.

For *TDMS* extraction, only five extracted *TDMS* tuples are correct. This is a challenging task and more efforts are required to address it in the future.

7 Zero-shot *TDM* Classification

Since our framework in principle captures the similarities between *DocTAET* and various *TDM* triples, we estimate that it can perform zero-shot classification of new *TDM* triples at test time.

We split *NLP-TDMS (Full)* into the training/test sets. The training set contains 210 papers with 96 (distinctive) *TDM* triple annotations and the test set contains 108 papers whose *TDM* triple annotations do not appear in the training set. We train our *DocTAET-TDM* model on the training set as described in Section 4.2.1. At test time, we use all valid *TDM* triples from *NLP-TDMS (Full)* to form the hypothesis space. To improve efficiency, one could also reduce this hypothesis space by focusing on the related *Task* or *Dataset* mentioned in the paper.

On the test set of zero-shot *TDM* pairs classification, our model achieves a macro F_1 score of 41.6 and a micro F_1 score of 54.9, versus the 56.6 macro F_1, and 66.0 micro F_1 of the few-shot *TDM* pairs classification described in Section 6.1.

8 Conclusions

In this paper, we have reported a framework to automatically extract tasks, datasets, evaluation metrics and scores from a set of published scientific papers in PDF format, in order to reconstruct the leaderboards for various tasks. We have proposed a method, inspired by natural language inference, to facilitate learning similarity patterns between labels and the content words of papers. Our first model extracts *<Task, Dataset, Metric>* (*TDM*) triples, and our second model associates the best score reported in the paper to the corresponding *TDM* triple. We created two datasets in the NLP domain to test our system. Experiments show that our model outperforms the baselines by a large margin in the identification of *TDM* triples.

In the future, more effort is needed to extract the best score. Also the work reported in this paper is based on a small *TDM* taxonomy, we plan to construct a *TDM* knowledge base and provide an applicable system for a wide range of NLP papers.

Acknowledgments

The authors appreciate the valuable feedback from the anonymous reviewers.

References

Waleed Ammar, Dirk Groeneveld, Chandra Bhagavatula, Iz Beltagy, Miles Crawford, Doug Downey, Jason Dunkelberger, Ahmed Elgohary, Sergey Feldman, Vu Ha, Rodney Kinney, Sebastian Kohlmeier, Kyle Lo, Tyler Murray, Hsu-Han Ooi, Matthew Peters, Joanna Power, Sam Skjonsberg, Lucy Wang, Chris Willhelm, Zheng Yuan, Madeleine van Zuylen, and Oren Etzioni. 2018. Construction of the literature graph in semantic scholar. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers),* New Orleans, Louisiana, 1 – 6 June 2018, pages 84–91.

Waleed Ammar, Matthew Peters, Chandra Bhagavatula, and Russell Power. 2017. The AI2 system at SemEval-2017 Task 10 (ScienceIE): Semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017),* Vancouver, Canada, 3 – 4 August 2017, pages 592–596.

Timothy G. Armstrong, Alistair Moffat, William Webber, and Justin Zobel. 2009. Improvements that don't add up: Ad-hoc retrieval results since 1998. In *Proceedings of the ACM 18th Conference on Information and Knowledge Management (CIKM 2009),* Hong Kong, China, 2–6 November 2009, pages 601–610.

Vaswani Ashish, Shazeer Noam, Parmar Niki, Uszkoreit Jakob, Jones Llion, Gomez Aidan N., Kaiser Lukasz, and Polosukhin Illia. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pages 1–11.

Awais Athar and Simone Teufel. 2012a. Context-enhanced citation sentiment detection. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Montréal, Québec, Canada, 3–8 June 2012, pages 597–601.

Awais Athar and Simone Teufel. 2012b. Detection of implicit citations for sentiment detection. In *Proceedings of the Workshop on Detecting Structure in Scholarly Discourse,* Jeju Island, Republic of Korea, 12 July, pages 18–26.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017),* Vancouver, Canada, 3 – 4 August 2017, pages 546–555.

Isabelle Augenstein and Anders Søgaard. 2017. Multi-task learning of keyphrase boundary classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* Vancouver, Canada, 30 July – 4 August 2017, pages 341–346.

Steven Bird, Robert Dale, Bonnie Dorr, Bryan Gibson, Mark Joseph, Min-Yen Kan, Dongwon Lee, Brett Powley, Dragomir Radev, and Yee Fan Tan. 2008. The ACL anthology reference corpus: A reference dataset for bibliographic research in computational linguistics. In *Proceedings of the 6th International Conference on Language Resources and Evaluation,* Marrakech, Morocco, 26 May – 1 June 2008, pages 1755–1759.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing,* Lisbon, Portugal, 17–21 September 2015, pages 632–642.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges*, pages 177–190, Heidelberg, Germany.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. 2014. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 1646–1654.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* Minneapolis, USA, 2–7 June 2019, pages 4171–4186.

Kathrin Eichler, Feiyu Xu, Hans Uszkoreit, and Sebastian Krause. 2017. Generating pattern-based entailment graphs for relation extraction. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017),* Vancouver, Canada, 3 – 4 August 2017, pages 220–229.

Nicola Ferro, Norbert Fuhr, and Andreas Rauber. 2018. Introduction to the special issue on reproducibility in information retrieval: Evaluation campaigns, collections, and analyses. *Journal of Data and Information Quality*, 10(3):9:1–9:4.

Kata Gábor, Davide Buscaldi, Anne-Kathrin Schumann, Behrang QasemiZadeh, Haïfa Zargayouna, and Thierry Charnois. 2018. Semeval-2018 task 7: Semantic relation extraction and classification in scientific papers. In *Proceedings of The 12th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT, New Orleans, Louisiana, June 5-6, 2018*, pages 679–688.

Sonal Gupta and Christopher Manning. 2011. Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of 5th international joint conference on natural language processing,* Chiang Mai, Thailand, 8–13 November 2011, pages 1–9.

Kata Gábor, Haifa Zargayouna, Davide Buscaldi, Isabelle Tellier, and Thierry Charnois. 2016. Semantic annotation of the ACL anthology corpus for the automatic analysis of scientific literature. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016),* Portorož, Slovenia, 23–28 May 2016.

David Jurgens, Srijan Kumar, Raine Hoover, Dan McFarland, and Dan Jurafsky. 2018. Measuring the evolution of a scientific field through citation frames. *Transactions of the Association for Computational Linguistics*, 6:391–406.

Seonhoon Kim, Jin-Hyuk Hong, Inho Kang, and Nojun Kwak. 2019. Semantic sentence matching with densely-connected recurrent and co-attentive information. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence,* Hawaii, USA, 27 January–1 February 2019.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics,* Florence, Italy, 28 July–2 August 2019.

Patrice Lopez. 2009. GROBID: combining automatic bibliographic data recognition and term extraction for scholarship publications. In *The 13th European Conference on Digital Libraries (ECDL 2009), Corfu, Greece, 27 September 27 – 2 October, 2009*, pages 473–474.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Brussels, Belgium, 31 October–4 November 2018, pages 3219–3232.

Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. 2017. Scientific information extraction with semi-supervised neural tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing,* Copenhagen, Denmark, 7–11 November 2017, pages 2641–2651.

Abiola Obamuyide and Andreas Vlachos. 2018. Zero-shot relation classification as textual entailment. In *Proceedings of the First Workshop on Fact Extraction and VERification (FEVER),* Brussels, Belgium, 1 November 2018, pages 72–78.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Adam Poliak, Aparajita Haldar, Rachel Rudinger, J. Edward Hu, Ellie Pavlick, Aaron Steven White, and Benjamin Van Durme. 2018. Collecting diverse natural language inference problems for sentence representation evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing,* Brussels, Belgium, 31 October– 4 November 2018, pages 67–81.

Chen-Tse Tsai, Gourab Kundu, and Dan Roth. 2013. Concept-based analysis of scientific literature. In *Proceedings of the ACM 22nd Conference on Information and Knowledge Management (CIKM 2013),* San Francisco, California, 27 October–1 November 2013, pages 1733–1738.

Adam Vogel and Dan Jurafsky. 2012. He said, she said: Gender in the acl anthology. In *Proceedings of the ACL-2012 Special Workshop on Rediscovering 50 Years of Discoveries,* Jeju Island, Republic of Korea, 10 July, pages 33–41.

Aaron Steven White, Pushpendre Rastogi, Kevin Duh, and Benjamin Van Durme. 2017. Inference is everything: Recasting semantic resources into a unified evaluation framework. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (IJCNLP 2017),* Taipei, Taiwan, 27 November – 1 December 2017, pages 996–1005.