Citation Intent Classification: Models Fusion and Reduction to Semantic Frame Classification

Anonymous ACL submission

Abstract

The scientific community has a growing interest on not considering all citations equal and on better characterising them. Deep learning techniques have been successfully exploited to classify citation intents, supported by public data-sets including SciCite (Cohan et al., 2019) and ACL-ARC (Jurgens et al., 2018). Our work shows a way to fuse models having different properties, in order to improve the overall performance of Deep Learning classifiers on Citation Intent Classification for a chosen domain. We created a new data-set called XXX-D[1], and we built on top of it a classifier by combining together existing models trained on similar problems. In particular our core idea is to reduce the Citation Intent Classification problem to that of Semantic Role Labeling and Semantic Frame Classification. Our approach has been evaluated with a 5-fold validation on XXX-D, achieving a mean F1-macro score of 84.77%.

1 Introduction

Citations are fundamental tools for researchers to link their results to other works. These links are mostly used as plain connections, simply counted for measuring impact or crossed to surf research areas. However the community is increasingly recognising the importance of not considering all citations equal (Ding et al., 2014), stressing out, for instance, the potentials for research evaluation purposes (Zhu et al., 2015).

Researchers have investigated automatic classification of citation intents with very good results (Hernández-Álvarez et al., 2016), (Jurgens et al., 2018). The most recent and successful approaches are based on deep learning techniques and use annotated citations to train classifiers and evaluate them (Cohan et al., 2019).

[1]Real name omitted for double blind review.

The amount of annotated data, as well as their quality, is obviously a key factor for the success of these approaches and their production is a critical and expensive task.

In this paper we present a new data-set, called XXX-D, that can be used for citation intent classification. Furthermore, we propose to use it in combination with existing models trained on problems similar to that at hand. Our basic idea, in fact, is that the citation intent classification problem can be *reduced* to the semantic frames classification (Bejan and Hathaway, 2007)(Hakkani-Tur et al., 2016). Thus, we experimented on reusing models trained for SFC in this new context.

We also used stacking techniques to test our approach in combination with two state-of-the-art models for citation classification intent, namely SciCite (Cohan et al., 2019) and ACL-ARC (Jurgens et al., 2018). Note that these models were trained to identify a different set of citation intents on different data-sets, but increased the performance of our classifier. This is indeed another form of reduction and gave us valuable indications about the feature selection process and its generality. Our approach has been evaluated with a 5-fold validation on XXX-D. We tested the results of different combinations of features, achieving a 84.77% f1-macro score with the best performing configuration.

The remaining of the paper is structured as follows. In Section 2 we discuss relevant citations' schemes and classifiers. Our pipeline is presented in Section 3, while Section 4 presents some details of the data-sets we used, in particular XXX-D. Experiments and results are presented in Section 5, before some lessons learned and our error analysis in Section 6.

2 Related Works

The research on classification of citation intents can be looked at from two perspectives: *citation intent models* and *automatic classifiers*.

Academics and researchers have proposed several citation intent models, ever since the seminal work of Garfield (1965). In the following we discuss the most relevant criteria to group them, and the characteristics along which these models have been differentiated. In Section 4 we will provide more details about the three models used for this project: XXX-D, SciCite and ACL-ARC.

The first aspect to take into account is the purpose of the classification. Despite some authors claim their models to be application independent (Jochim and Schütze, 2012), the lack of an univocal model is the result of a diversity of applications and a major hindrance to a fair comparison of the related procedures. The purpose may vary from theoretical social studies (Harwood, 2009) to concrete applications (Nanba and Okumura, 1999) where polarity-like (Chubin and Moitra, 1975) and value-like dimensions (Moravcsik, 1973) are studied.

Models differ also in the degree of details which is another purpose related characteristic. From a handful of classes (Dong and Schafer, 2011) to more than forty categories (Peroni and Shotton, 2012), the model can be shaped on the classification needed for the task.

The scientific domain of the paper is another factor that influences the structure of a model. Citation function models developed for specific domains show peculiarities as the relevance of identifying used methodologies, data, techniques etc. in scientific domain (e.g. (Teufel et al., 2006)), or determining theories and opinions in humanities (e.g. (Frost, 1979)).

Most citation intent classifiers are conceived, or at least trained, on the computational linguistic domain. They use citation intent models based on past works, opting for a coarse granularity (around 4-6 classes) in order to sharpen the task.

The features on which classifiers are based are differently extensive but can be mainly classified into: **structural features** (location of the citation, section type, density of occurrence etc), **lexical features** (cue words, connective phrases, pronouns etc), **linguistic features** (verb tenses, POS tags, dependencies etc) and **cited entity features** (self citations, paper topic, venues details etc).

The authors usually explored the capability of current state of the art classifier algorithms, and good performance are obtained with a variety of them (BayesNet, IBK from the Weka toolkit, Random Forest, Stanford MaxEnt classifier, SVM).

The two most relevant works for our research are SciCite (Cohan et al., 2019) and ACL-ARC (Jurgens et al., 2018) classifiers. The first one exploits a "scaffold" neural model to incorporate structural information of scientific papers - for instance, some information about the section containing the citation and the 'citation worthiness' - into citation. Trained on very large data-set of about 11000 citation, publicly released and described in Section 4, the model showed high accuracy with a f1-macro score of 84%. SciCite outperformed the ACL-ARC classifier that achieved a f1-macro score of 51.8%. ACL-ARC was trained on a smaller dataset of about 2000 annotated citations, but was able to use a larger model of six citation intents we discuss in the next Sections.

3 Our Approach

Our work shows a way to fuse data-sets having different properties, by stacking together pre-trained models trained on these data-sets. In order to do it effectively, we propose a way to identify which data-sets are suitable or not, and why. Our goal is to improve the overall performance of Deep Learning classifiers on Citation Intent Classification (CIC), for a chosen domain. Our core idea is to reduce the CIC problem to that of Semantic Role Labeling (SRL) and Semantic Frame Classification (SFC).

3.1 The Overall Idea of Reduction

Currently, Deep Learning (DL) seems to be one of the most effective tools we have to Natural Language Processing (NLP). DL usually requires a considerable amount of data to properly work. This is why several techniques for transfer learning (Ruder et al., 2019; Artetxe et al., 2017) and n-shot learning (Srivastava et al., 2018; Yazdani and Henderson, 2015) have been proposed by the community.

The process of generating labeled data-sets of citation intents is complex, expensive and time-consuming. This was also observed in (Cohan et al., 2019) and (Jurgens et al., 2018), which presented respectively SciCite and ACL-ARC and worked on different sets of citation intents.

The direct application of SciCite or ACL-ARC models to our context was not viable, mainly because their citation intent categories are different

from ours. Considering the fact that for DL we would need a couple of orders of magnitude more samples than those in XXX-D, a further manual annotation process to increase XXX-D was not feasible either. We came up with a different solution, based on the idea of **problem reduction**, coming from computational complexity.

We believe that CIC is reducible to the more general problems of Semantic Frame Classification (SFC) and Parsing (SFP), Semantic Role Labeling (SRL) and Semantic Retrieval (SR). This implies that an algorithm for SFC, SFP, SRL or SR could be easily adapted and incorporated as sub-routine, to work for CIC.

In fact, being able to identify the citation intent in a sentence requires to understand the semantics of the sentence, locating the elements in the sentence that evoke a particular intent. These elements, evoking the intent, assume a specific role in the sentence. For example, in a sentence expressing the citation intent "Uses Method In" there are going to be one or more entities having the role of: user, method, recipient, and/or container. Intuitively, in order to properly identify a "Uses Method In" intent we would have to identify in a sentence one or more of the aforementioned semantic roles. So, being able to perform Semantic Role Labeling is a step for Citation Intent Classification, as much as Semantic Frame Parsing and Semantic Retrieval. In fact SR has been shown to be useful for Question Answering (QA) (Yang et al., 2019), because in order to be able to answer a question, we have first of all to identify the intent of the question and the role of the entities in the question, that is a task involving SRL.

Many algorithms are known for SFP and SRL, including: FRED (Gangemi et al., 2017) (an expert system), Open-Sesame (Swayamdipta et al., 2017) (based on Recurrent Neural Networks), SEMAPHORE (Kshirsagar et al., 2015) (based on log-linear models), SLING (Ringgaard et al., 2017) (based on Recurrent Neural Networks). While for SR+QA we mention the Universal Sentence Encoder (USE) (Yang et al., 2019).

If we want to effectively exploit pre-trained deep learning models for citation intent classification, what we have to do is to identify one or more DL algorithm for SFP, SRL and/or SR and use them as sub-routine of our citation intent classifier. Due to the simplicity of its APIs, we decided to adopt USE as default algorithm for our reduction. Using

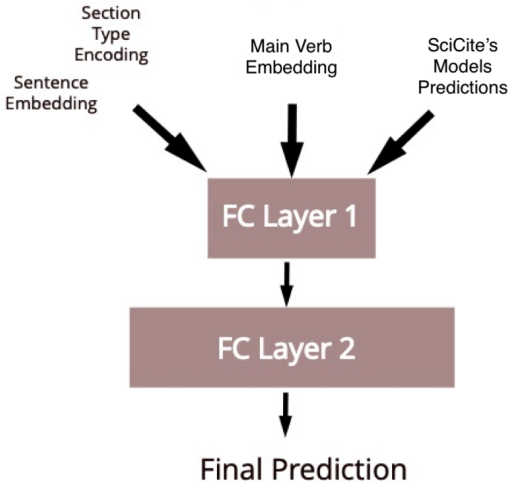

Figure 1: Overview of our stacking architecture

USE we are able to encode every sentence into a vector having the fixed length of 512 features.

So, what we would expect is that, the more a problem P is similar to CIC, the more a DL model for tackling P is going to be useful for tackling CIC. According to this, we would expect that:

- Using the QA version of USE (trained to tackle SR) is going to be more effective than using the standard version of USE (trained to tackle Semantic Similarity, that is more generic than SR).
- Stacking other models, generic-enough and trained to solve similar CIC problems, is going to improve the overall performances of our classifier.

The SciCite (Cohan et al., 2019) pre-trained models are suitable candidates to verify our hypothesis. These models are two different models in the sense that they have been trained on two different data-sets: the ACL-ARC data-set and the SciCite data-set.

To reinforce the validity of our hypothesis that CIC is reducible to SRL, following the intuition that the semantic roles are usually defined by predicates, we also expect that showing to the classifier extra features about these predicates (easily identifiable through a dependency parser) might be useful to make our classifier focus on the most difficult part of SRL: the identification of the semantic roles. This is why we extract from every sentence its main predicate in order to embed it with USE and feed it as extra feature to the classifier. We defined the main predicate as the predicate having the biggest

number of different Subject-Verb-Object (SVO) triples involving it. For example, according to our SVO extraction algorithm, the sentence "A person can be a natural entity or a legal entity, and cannot be an object" contains two predicates and expanded as four SVO triples: (1) person can be natural entity, (2) person can be legal entity, (3) person can be entity and (4) person cannot be object. In this example, the first predicate "can be" has three SVO triples, while the second predicate only one, so the first predicate is said to be the main verb of the sentence.

3.2 SciCite Classifier and Model

In figure 1 we show the stacking architecture we adopted. We may have different types of features:

- The sentence embedding.
- The main verb embedding.
- The encoding of the section category containing the sentence.
- The categorical distribution predicted by pre-trained models such as SciCite's ones.

The neural architecture adopted for verifying our hypothesis is shown in figure 1 and it is made of an input layer, followed by a Fully-Connected (FC) having a non-linear activation function, followed by another FC layer as output layer having a softmax as activation function. For more details about the hyper-parameters and the experiments, please read Section 5. The code is open-source and available at XX.[2]

4 Data-sets

Three data-sets were used in our experiments, all described in this Section.

4.1 XXX-D

XXX-D is a set of 1370 annotated citations available at XX[3]. It contains, for each citation, the sentence including the citation (i.e. citation context), the full bibliographic reference, the title of the section in which the paper is cited, and the human-annotated citation intent. The data-set was developed as part of a pre-competitive project in partnership with one of the biggest scientific publisher and academic platforms provider. Although the full project was meant to cover any scientific domain, we restricted our experiments to the field of Computer Science. Preliminary experiments in Chemistry highlighted several difficulties in agreeing about citation functions in a domain we and the annotators were not expert of (as computer scientists). Hence we collected a selection of 14380 articles from 160 different journals and published between 2010 and 2017, and extracted some of their citations.

Our focus was on widespread factual intents[4]. The initial set of potentially relevant intents was validated with an open survey, to which 321 researchers responded from different countries. We selected four of them with the goal of minimizing the effort of the annotators and producing high-quality data for automatic classification.

Table 2 summarizes XXX-D intent categories together with the two other data-sets we took into account. We considered our categories plus one residual class: 1) *uses method in*, 2) *extends*, 3) *uses data from*, 4) *cites as review*, and 5) *cites* when no other functions can be assigned.

Data-set	Domain	Annotators	# Cit.	Intent details
XXX-D	Computer Science	3 experts	1370	Uses method in (28%) Uses data from (11%) Extends (6%) Cites (55%)
SciCite	Computer Science — Medicine	850 crowdsourced	11020	Background (58%) Method (29%) Compare results (13%)
ACL-ARC	Computational Linguistics	2 experts	1941	Background (51%) Extends (4%) Uses (19%) Motivation (5%) Compare (18%) Future Work (4%)

Table 1: Overview of XXX-D, SciCite and ACL-ARC datasets.

Intent	Examples
Uses_method_in	All raw NMR data from HSQC-TOCSY experiments were processed with nmrPipe software [32] applying Gaussian multiplication in both dimensions.
Uses_data_from	We mainly experiment on the Semantic 3D dataset [40]
Extends	The current paper extends the previous one [5] along numerous axes.
Cites	As the rapid development of the web technology, in particular in the social network, there emerge some multi-view graphs [14].

Table 2: Examples of the citation intents in XXX-D.

[2]Omitted for blind review, will be made available. Temporarily uploaded as supplementary material.

[3]Omitted for blind review, will be made available. Temporarily uploaded as supplementary material.

[4]We also use the term *function* in the paper. The two terms are interchangeable.

The identification of these functions did not happen in one single step but was the result of an incremental process. A small group of internal experts (three in total) collected examples and counterexamples of each function and merged them into a set of guidelines and explanatory annotations.

For example, note that existing models do not make distinction between the use of data and the use of methods which are usually merged in a single category (Jurgens et al., 2018). We kept them as two distinct classes for a better characterization.

The annotators also produced in full agreement a first set of 820 annotated citations. The extraction of citation data to be annotated has been done oversampling some citation contexts. Some articles from the corpus were randomly chosen and all citation contexts extracted, taking the same amount of contexts with and without some cue words, so as to balance biased and unbiased samples.

To expand training data, we exploited the citation network selecting other citations towards papers already cited in the first set of citations. Three other annotators, supported by a prototype classifier, manually compared the functions assigned to these new citations to those already available in the training data-set and pointing to the same paper. We then filtered only those for which the latter were equal to the predicted ones and we got 146 more validated annotated citations.

The annotators also annotated other 584 randomly-chosen citations starting from output of the classifier, so as to feed the data-set with more false positives. In conclusion, the data set comprises 1370 different citations.

4.2 ACL-ARC and SciCite

SciCite is a large and cross-domain set of 11020 annotated citations. As we did for XXX-D, SciCite authors selected a limited set of intent categories, with the goal of reducing annotators effort and classifiers errors. They aggregated fine-grained intents into three classes: *background*, *method* and *result comparison*. Actually a fourth residual class was initially included (called 'other' and comparable to the XXX-D 'cite') but then was discarded since it was used only a few times during the annotation process.

Instead of being annotated by internal experts, SciCite data were primarily labelled through a crowd-sourcing platform[5]. The agreement between annotators was used as confidence score and annotations with agreement of <0.7 were discarded. A final pass by an expert annotator on some samples was deployed to further check quality.

Citations were taken in the fields of computer science and medicine and annotated by 850 people, with an average of 3.74 per citation. The total counts 9,159 annotated citations, plus 1,861 annotated by a trained annotator.

ACL-ARC is smaller but classifies a larger number of intents (Jurgens et al., 2018). Six classes were used: *background*, *extends*, *uses*, *motivation*, *compare* and *future work*.

The articles domain is also different from XXX-D and SciCite, being focused on the linguistic area, and including citations taken from papers in the ACL Anthology Reference Corpus.

ACL-ARC was built in two phases by two internal experts in NLP. They initially worked on 10 papers and agreed on the guidelines and the meaning of the citation intents. Then, they annotated citations randomly extracted from 185 papers, for a total of 1969 annotated citations. Table 1 summarizes the overall structure, distribution of classes and features of ACL-ARC in comparison to SciCite and XXX-D.

5 Experiments

Six different experiments were performed. All the experiments adopt the same neural architecture, the only difference between the experiments are the features used as input. The experiments are the following:

1. **Transformer**: we use as features only the Section Type Encoding (STE) and the Sentence Embedding (SE) obtained through the standard version of USE (Cer et al., 2018), not trained to perform Semantic Retrieval.

2. **MLQA**: we use as features the STE and the SE obtained through the Multi-Lingual Question-Answering (MLQA) version of USE (Yang et al., 2019), trained to perform Semantic Retrieval.

3. **MLQA + Main Verb**: we use as features the STE, the SE obtained through MLQA, and the Main Verb (as described in Section 3).

4. **MLQA + Main Verb + SciCite data-set**: we use as features the STE, the SE obtained through MLQA, the Main Verb (MV), and

[5]FigureEight, https://www.figure-eight.com/

the categorical distributions predicted by SciCite's model trained on SciCite data-set.

5. **MLQA + Main Verb + ACL-ARC data-set**: we use as features the STE, the SE obtained through MLQA, the MV, and the categorical distributions predicted by SciCite's model trained on ACL-ARC data-set.

6. **MLQA + Main Verb + SciCite and ACL-ARC data-sets**: we use as features the STE, the SE obtained through MLQA, the MV, and the categorical distributions predicted by SciCite's models trained on SciCite and ACL-ARC data-sets.

5.1 Goals of the Experiments

Experiments 1 and 2 are meant to sustain the hypothesis that the more a problem P is similar to CIC, the more a DL model for tackling P is going to be useful for tackling CIC. In fact MLQA has been trained to solve Semantic Retrieval that is a more similar problem to CIC than the more generic Semantic Similarity problem. While experiments 3 and 2 are meant to sustain the hypothesis that CIC is reducible to Semantic Role Labeling. The remaining experiments are used to show that stacking other models, generic-enough and trained to solve similar CIC problems, is going to improve the overall performances of our classifier and its generality, thus reducing over-fitting.

5.2 Hyper-Parameters Tuning

The neural architecture adopted for the experiments is shown in figure 1, and it is made of an input layer, followed by a Fully-Connected (FC) layer of U units and having F as activation function, followed by another FC layer as output layer having a softmax as activation function. The gradient optimizer we adopted is Proximal Adagrad with learning rate L and L2 regularization strength R.

Due to the fact the XXX-D data-set is relatively small, we adopted more or less all the known tricks for preventing over-fitting with neural networks:

- L2 regularization (directly on the gradient optimization).
- A small batch size B.
- Data Augmentation, through re-sampling (over- and/or under- sampling) algorithms S.

Here, B, S, U, F, L and R are hyper-parameters we automatically tune. We performed automatic tuning of hyper-parameters through Grid Search and the scheduling algorithm Asynchronous Hyperband (Liaw et al., 2018). The following hyper-parameters were tested, for a total of 1800 different combinations per experiment:

- Units U: 4, 8, 12.
- Activation Function F: RELU, SELU, TANH.
- Learning rate L: 0.3, 0.1, 0.03, 0.01.
- Regularization strength R: 0.01, 0.003, 0.001, 0.0003, 0.0001.
- Re-sampling algorithm S: None (no re-sampling), SMOTE-Tomek (Batista et al., 2003), SMOTE (Chawla et al., 2002), ADASYN (He et al., 2008), TomekLinks (Tomek, 1976).
- Batch size B: 2, 4.

For every experiment we performed a 5-fold cross validation. We forced all experiments to have the same test-set and training-set splits (by fixing the seed for the generation of pseudo-random numbers), thus being able to better compare the results.

5.3 Baseline

Our baseline is the SciCite's model specifically trained on our data-set. The result we got is a F1-macro with mean 78.58% and standard deviation 2.95%. We think that we were not able to get very good results, training the SciCite's model directly on our data-set, mainly because our data-set is very small and imbalanced and not suited for deep learning models such as SciCite's one.

6 Results and Discussion

The results of the experiments are shown in table 3. As we can see, our classifier achieves the best performances (F1-Macro score with mean 84.77% and standard deviation 1.43%) by using together: the MLQA sentence embedding, the Main Verb embedding, and both the SciCite's pre-trained models. The results somehow confirm our main hypothesis that the more a problem P is similar to Citation Intent Classification (CIC), the more a DL model for tackling P is going to be useful for tackling CIC.

Experiment	F1-macro mean	F1-Macro std	F1-micro mean	F1-micro std
MLQA + MV + SciCite + ACL-ARC	84.77%	1.43%	84.74%	1.59%
MLQA + MV + ACL-ARC	83.64%	1.39%	83.81%	0.95%
MLQA + MV + SciCite	82.61%	1.25%	83.74%	1.03%
MLQA + MV	82.62%	2.57%	82.94%	1.26%
MLQA	81.42%	2.34%	82.26%	1.68%
Transformer	78.81%	3.18%	80.78%	2.59%

Table 3: Experiments Results

Figure 2: Normalized Confusion Matrix of MLQA + MV + SciCite + ACL-ARC.

In Figure 2 we show the normalized confusion matrix on the test-set of the 4th fold of experiment 6. As we can see, the accuracy is quite homogeneous in all the classes, despite the fact that the class distribution in XXX-D is highly imbalanced toward the classes "cites" and "uses method in". Despite this, surprisingly, the best accuracy is achieved for the class "uses data from", and it is around 90%.

Figure 3: Normalized Confusion Matrix of Transformer.

Just for sake of comparison, in Figure 3 we show the normalized confusion matrix of experiment 1. Experiment 1 does not exploit any pre-trained SciCite's model, and as expected we can see that it performs poorly on both the under-represented classes of XXX-D data-set. Comparing these two normalized confusion matrices we are able to see that our reduction and stacking technique is able to improve the overall generalization, significantly mitigating the imbalancedness problem.

Furthermore, other interesting results on the automatic hyper-parameters tuning are:

- In the Transformer experiment, re-sampling seems to be not effective.
- The smallest batch size seems to perform better.
- SMOTE is the most effective re-sampling technique. Even SMOTE-Tomek produces good results but not in every experiment.
- A greater number of units on average gives better results (e.g. $U = 12$ beats $U = 4$).
- SELU seems to be the best activation function, followed by RELU and TANH.
- The best learning rate is close to 0.1, sometimes even 0.3.
- In the case of the Transformer experiment, the optimal regularization coefficient is 0.003, one order of magnitude greater than in the other experiments, for which the best coefficient is 0.0003.

These results confirm that adopting the classical techniques for mitigating over-fitting is a good strategy, as expected. Furthermore, the interesting findings on the regularization coefficients confirm, again, that our reduction approach is helping the classifier to significantly reduce over-fitting.

6.1 Error Analysis

We performed an error analysis by comparing the errors in the test-set of the 4th fold of the best hyper-parameters configuration of every experiment. We took the 4th fold because it appeared to be nor the

Sample errors on 'uses method in'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
In order to validate the effectiveness of our algorithm DyTrust, we conducted extensive experiments with the real trust network dataset, Kaitiaki and compared the results of DyTrust with those of the well-known with trust propagation algorithms, as well as with those of our previous algorithms on trust propagation [9].	In conducted	88.86%	**94.07%**	0.46%	0.25%	5.20%
For state of the art algorithms, if we apply the Spectral Regression Discriminant Analysis algorithm in [80], the time complexity of LDA can reach O(NSm).	For can reach	2.88%	**51.39%**	0.02%	0.06%	48.51%

Sample errors on 'uses data from'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
In terms of diversity, the corpus consists of RDF from 783 pay-level-domains [32]	In consists of	76.48%	**76.75%**	0.04%	0.27%	22.93%
We compared the data of this study with eye tracking data of the memorability experiment [2] with conditions closer to natural image viewing.	compared	0.80%	**50.16%**	0.04%	49.35%	0.43%

Sample errors on 'extends'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
Our system extends the technique proposed in [13] to represent and manipulate cloth simulation data as well.	extends	51.52%	**75.59%**	24.06%	0.00%	0.33%
Our approach extends the RankJoin operator in [8] to work with UNCQs.	extends in	31.38%	10.09%	29.23%	0.05%	**60.61%**

Sample errors on 'cites'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
For HOFM, we use the parameter settings as per [9] where a regular grid of size 30 30 3 is created.	For use as	97.20%	1.25%	0.00%	0.28%	**98.46%**
Such sampling technique provides good sources of data in exploratory research (Easterby-Smith, Thorpe, & Jackson, 2015)	provides	0.84%	46.47%	0.06%	**47.31%**	6.14%

Table 4: Some sample errors and errors deltas in our best experiment (MLQA + MV + SciCite + ACL-ARC). Cells in bold indicate the label assigned by the classifier.

one giving the best results, nor the one giving the worst results.

During our error analysis we took under considerations the quantity and the quality of the errors committed by our algorithm. We grouped by (true) label the samples wrongly classified, and then we sorted them by the *error delta*. What is the *error delta*? For every sample, the classifier computes a categorical distribution, that is a vector of length C where C is the number of the different citation intents to classify (for SCAR-D $C = 4$). Let p_w be the highest probability in the aforementioned categorical distribution, and let p_c be the probability assigned to the correct class. The *error delta* is the difference between p_w and p_c. In the case of the classifier committing a mistake, p_w is always greater than p_c. Thus, the *error delta* shows how much strong is the classifier error.

We define as *weak errors* those errors having *error delta* lower than 10%. Weak errors are due to ambiguity perceived by the algorithm in the text. In other words, when the algorithm is not sure whether the outcome should be the labeled class or another one, it gives as outcome similar probabilities for both of them, thus generating a weak error.

In table 5 we show the strongest and the weakest errors for each class on our best experiment, while, in the following sub-sections, we discuss the error analysis of all the six experiments.

6.1.1 Experiment 1: Transformer

In this experiment, the words similar to "data" frequently trigger "uses data from" as outcome, wrongly. On the other hand, apparently there is no clear error pattern for false positive on classes "uses method in" and "extends".

6.1.2 Experiment 2: Semantic Retrieval

In this experiment, the amount of errors for class "uses data from" is drastically lower. Furthermore, we can see more error patterns related to the presence of keywords in the sentences to classify. Additionally to the error patterns identified in experiment 1, we have that:

- The words similar to "extend" are likely to trigger "extends" as outcome, wrongly.
- Words similar to "use" are likely to trigger "uses method in" as outcome. This behaviour happens just in a few cases, not all of them.

In the case of "cites" intents wrongly classified by the algorithm, the 18.75% of the errors are weak. Results show that classes "uses method in" and "cites" have the highest percentage of errors in the training set, while in the test set the most mistaken classes are "extends" and "uses method in".

Sample errors on 'uses method in'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
In order to validate the effectiveness of our algorithm DyTrust, we conducted extensive experiments with the real trust network dataset, Kaitiaki and compared the results of DyTrust with those of the well-known with trust propagation algorithms, as well as with those of our previous algorithms on trust propagation [9].	In conducted	88.86%	**94.07%**	0.46%	0.25%	5.20%
For state of the art algorithms, if we apply the Spectral Regression Discriminant Analysis algorithm in [80], the time complexity of LDA can reach O(NSm).	For can reach	2.88%	**51.39%**	0.02%	0.06%	48.51%

Sample errors on 'uses data from'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
In terms of diversity, the corpus consists of RDF from 783 pay-level-domains [32]	In consists of	76.48%	**76.75%**	0.04%	0.27%	22.93%
We compared the data of this study with eye tracking data of the memorability experiment [2] with conditions closer to natural image viewing.	compared	0.80%	**50.16%**	0.04%	49.35%	0.43%

Sample errors on 'extends'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
Our system extends the technique proposed in [13] to represent and manipulate cloth simulation data as well.	extends	51.52%	**75.59%**	24.06%	0.00%	0.33%
Our approach extends the RankJoin operator in [8] to work with UNCQs.	extends in	31.38%	10.09%	29.23%	0.05%	**60.61%**

Sample errors on 'cites'

text	*main_predicate*	*error_delta*	*cites*	*extends*	*uses_data_from*	*uses_method_in*
For HOFM, we use the parameter settings as per [9] where a regular grid of size 30 30 3 is created.	For use as	97.20%	1.25%	0.00%	0.28%	**98.46%**
Such sampling technique provides good sources of data in exploratory research (Easterby-Smith, Thorpe, & Jackson, 2015)	provides	0.84%	46.47%	0.06%	**47.31%**	6.14%

Table 5: Some sample errors and errors deltas. Cells in bold indicate the label assigned by the classifier.

6.1.3 Experiment 3: Main Verb

In this experiment, we can see that using the main verbs as extra features reduces the amount errors for the class "uses method in", but it increases the other error percentages. This is because, now, words such as "extend" and "use" have more importance, thus generating more false positives respectively for classes "extends" and "uses method in".

6.1.4 Experiment 4: SciCite Data-set

In this experiment, we can see that stacking with the SciCite data-set helps reducing, drastically, the amount of false positive for the class "uses method in", somehow overcoming the problem arisen by introducing the Main Verbs as extra feature. Now, the percentage of local errors for the class "cites" is significantly lower, going from around 15% to around 10%.

6.1.5 Experiment 5: ACL-ARC Data-set

In this experiment, we can see that using the ACL-ARC data-set drastically reduces the false positives for the class "extends". This experiment performs significantly worse than experiment 4 for the class "uses method in".

6.1.6 Experiment 6: All in

This experiment makes more mistakes on the training set, but it seems to generalize better by giving better results on the test-set, in average. Furthermore, in the case of "uses data from" intents wrongly classified by the algorithm, the 33.33% of the errors is weak. In other words, the combination of both SciCite and ACL-ARC seems to provide weaker errors on the identification of "uses data from" intents.

7 Conclusions

The goal of this paper was to further investigate the application of Deep Learning approaches to the classification of citation intents (CIC) in real-world applications.

Here we presented a new data-set called XXX-D and we applied the idea of problem reduction in order to exploit pre-trained models, even built in different contexts and for different problems, to our case.

Reducing CIC to Semantic Frame Classification (SFC) and Semantic Retrieval (SR), in fact, we managed to build a high performing classifier, further improved by stacking state-of-the-art models for different sets of citation intents.

We hypothesize that stacking with more models, solving problems similar to CIC, can improve the overall performances. New Citation Intent data-sets properly crafted to be different from XXX-D, SciCite, and ACL-ARC might be used to increase

the overall F1-macro.

Conversely XXX-D could be used to train other models and to solve different problems, an area we plan to investigate in the near future and that could open interesting and challenging opportunities for the community.

References

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2017. Learning bilingual word embeddings with (almost) no bilingual data. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462.

Gustavo EAPA Batista, Ana LC Bazzan, and Maria Carolina Monard. 2003. Balancing training data for automated annotation of keywords: a case study. In *WOB*, pages 10–18.

Cosmin Adrian Bejan and Chris Hathaway. 2007. UTD-SRL: A pipeline architecture for extracting frame semantic structures. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 460–463, Prague, Czech Republic. Association for Computational Linguistics.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Daryl E. Chubin and Soumyo D. Moitra. 1975. Content Analysis of References: Adjunct or Alternative to Citation Counting?

Arman Cohan, Waleed Ammar, Madeleine van Zuylen, and Field Cady. 2019. Structural scaffolds for citation intent classification in scientific publications. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3586–3596, Minneapolis, Minnesota. Association for Computational Linguistics.

Ying Ding, Guo Zhang, Tamy Chambers, Min Song, Xiaolong Wang, and Chengxiang Zhai. 2014. Content-based citation analysis: The next generation of citation analysis. *J. Assoc. Inf. Sci. Technol.*, 65(9):1820–1833.

Cailing Dong and Ulrich Schafer. 2011. Ensemble-style Self-training on Citation Classification. *Language*.

Carolyn O. Frost. 1979. The Use of Citations in Literary Research: A Preliminary Classification of Citation Functions. *The Library Quarterly: Information, Community, Policy*, 49(4):399–414.

Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovì. 2017. Semantic web machine reading with fred. *Semantic Web*, 8(6):873–893.

Eugene Garfield. 1965. Can Citation Indexing be Automated? In *Statistical Assoc . Methods for Mechanized Documentation*.

Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of Interspeech*.

Nigel Harwood. 2009. An interview-based study of the functions of citations in academic writing across two disciplines. *Journal of Pragmatics*.

Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. IEEE.

Myriam Hernández-Álvarez, José Gómez Soriano, and Patricio Martínez-Barco. 2016. Annotated Corpus for Citation Context Analysis. *Latin American Journal of Computing Faculty of Systems Engineering National Polytechnic School Quito-Ecuador*.

Charles Jochim and Hinrich Schütze. 2012. Towards a Generic and Flexible Citation Classifier Based on a Faceted Classification Scheme. *Procedings of COLING'12*.

David Jurgens, Raine Hoover, and Dan Mcfarland. 2018. Measuring the Evolution of a Scientific Field through Citation Frames. *TACL*.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime G Carbonell, Noah A Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 218–224.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Michael J Moravcsik. 1973. Measures of scientific growth. *Research Policy*, 2(3):266–275.

Hidetsugu Nanba and Manabu Okumura. 1999. Towards multi-paper summarization using reference information. In *IJCAI International Joint Conference on Artificial Intelligence*.

Silvio Peroni and David Shotton. 2012. FaBiO and CiTO: Ontologies for describing bibliographic resources and citations.

Michael Ringgaard, Rahul Gupta, and Fernando CN Pereira. 2017. Sling: A framework for frame semantic parsing. *arXiv preprint arXiv:1710.07032*.

Sebastian Ruder, Matthew E Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18.

Shashank Srivastava, Igor Labutov, and Tom Mitchell. 2018. Zero-shot learning of classifiers from natural language quantification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 306–316.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. *arXiv preprint arXiv:1706.09528*.

Simone Teufel, Advaith Siddharthan, and Dan Tidhar. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing - EMNLP '06*.

Ivan Tomek. 1976. Two modifications of cnn. *IEEE Trans. Systems, Man and Cybernetics*, 6:769–772.

Yinfei Yang, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, et al. 2019. Multilingual universal sentence encoder for semantic retrieval. *arXiv preprint arXiv:1907.04307*.

Majid Yazdani and James Henderson. 2015. A model of zero-shot learning of spoken language understanding. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 244–249.

Xiaodan Zhu, Peter Turney, Daniel Lemire, and André Vellino. 2015. Measuring academic influence: Not all citations are equal. *Journal of the Association for Information Science and Technology*, 66(2):408–427.