Rhetorical Classification of Anchor Text for Citation Recommendation

Daniel Duma
University of Edinburgh
danielduma@gmail.com

Maria Liakata
University of Warwick
M.Liakata@warwick.ac.uk

Amanda Clare
Aberystwyth University
afc@aber.ac.uk

James Ravenscroft
University of Warwick
ravenscroft@papro.org.uk

Ewan Klein
University of Edinburgh
ewan@inf.ed.ac.uk

CCS Concepts

•**Computing methodologies** → **Natural language processing;** *Discourse, dialogue and pragmatics;* •**Information systems** → *Information retrieval;*

Keywords

Core Scientific Concepts; CoreSC; context based; citation recommendation; anchor text; incoming link contexts

ABSTRACT

Wouldn't it be helpful if your text editor automatically suggested papers that are contextually relevant to your work? We concern ourselves with this task: we desire to recommend contextually relevant citations to the author of a paper. A number of rhetorical annotation schemes for academic articles have been developed over the years, and it has often been suggested that they could find application in Information Retrieval scenarios such as this one. In this paper we investigate the usefulness for this task of CoreSC, a sentence-based, functional, scientific discourse annotation scheme (e.g. Hypothesis, Method, Result, etc.). We specifically apply this to anchor text, that is, the text surrounding a citation, which is an important source of data for building document representations. By annotating each sentence in every document with CoreSC and indexing them separately by sentence class, we aim to build a more useful vector-space representation of documents in our collection. Our results show consistent links between types of citing sentences and types of cited sentences in anchor text, which we argue can indeed be exploited to increase the relevance of recommendations.

1. INTRODUCTION

Scientific papers follow a formal structure, and the language of academia requires clear argumentation [9]. This has led to the creation of classification schemes for the rhetorical and argumentative structure of scientific papers, of which two of the most prominent are Argumenative Zoning [19] and Core Scientific Concepts (CoreSC, [11]). The former focusses on the relation between current and previous work whereas the latter mostly on the content of a scientific investigation. These are among the first approaches to incorporate successful automatic classification of sentences in full scientific papers, using a supervised machine learning approach.

It has often been suggested that these rhetorical schemes could be applied in information retrieval scenarios ([19], [12], [3]). Indeed, some experimental academic retrieval tools have tried applying them to different retrieval modes ([18], [14], [1]), and here we explore their potential application to a deeper integration with the writing process.

Our aim is to make automatic citation recommendation as relevant as possible to the author's needs and to integrate it into the authoring workflow. Automatically recommending contextually relevant academic literature can help the author identify relevant previous work and find contrasting methods and results. In this work we specifically look at the domain of biomedical science, and examine the usefulness of CoreSC for this purpose.

2. PREVIOUS WORK

The ever-increasing volume of scientific literature is a fact, and the need to navigate it a real one. This has brought much attention to the task of *Context-Based Citation Recommendation* (CBCR) over the last few years [6, 5, 3, 7]. The task consists in recommending relevant papers to be cited at a specific point in a draft scientific paper, and is universally framed as an information retrieval scenario.

We need to recommend a citation for each *citation placeholder*: a special token inserted in the text of a draft paper where the citation should appear. In a standard IR approach, the corpus of potential papers to recommend (the *document collection*) is indexed for retrieval using a standard vector-space-model approach. Then, for each citation placeholder, the *query* is generated from the textual context around it (the *citing context*), and a *similarity* measure between the query and each document is then applied to rank the documents in the collection. A list of documents ranked by relevance is returned in reply to the query, so as to maximise the chance of picking the most useful paper to cite.

WOSP '16 22–23 June, 2016, Newark, NJ, USA

ACM ISBN .

DOI:

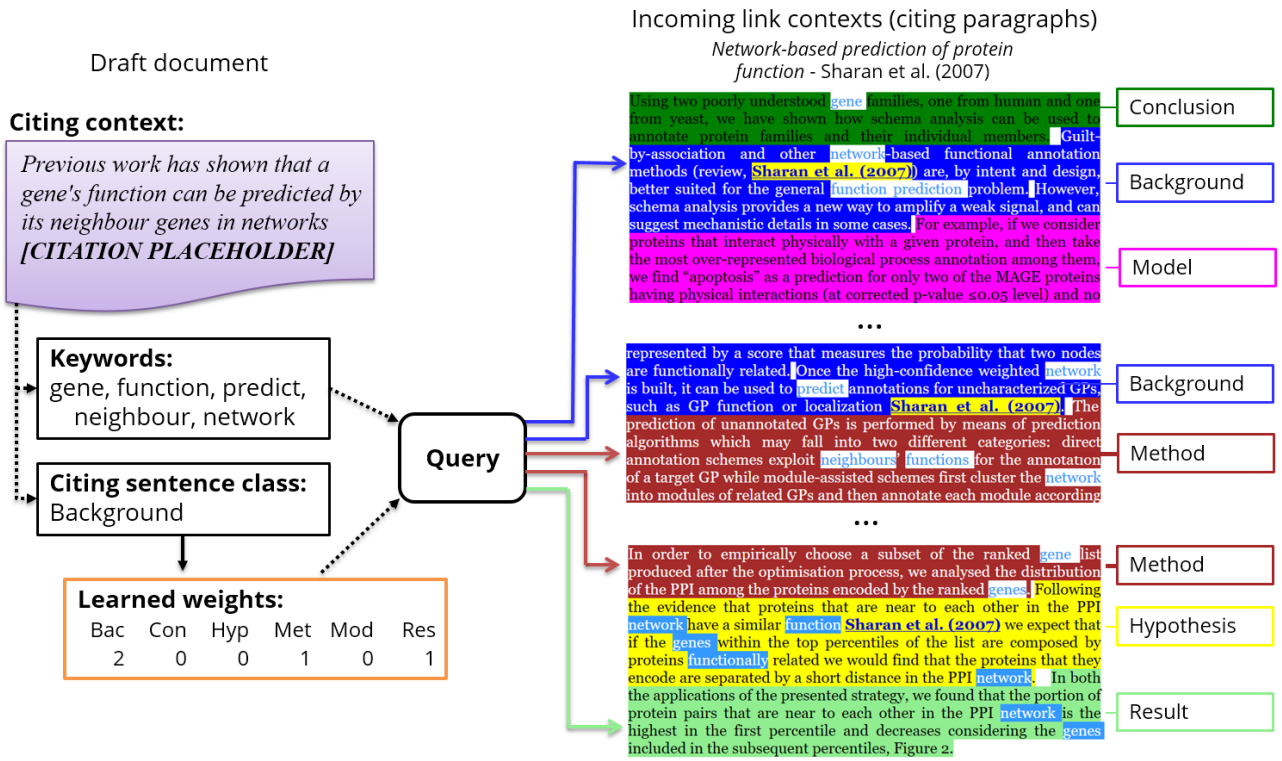

Figure 1: A high-level illustration of our approach. The class of the citing sentence is the *query type* and it determines a set of weights to apply to the classes of sentences in the anchor paragraphs of links to documents in our collection. In this example, for Bac, only 3 classes have non-zero weights: Bac, Met and Res. We show extracts from 3 different citing papers, exemplifying terms matching in different classes of sentences.

The *citing sentence* is the sentence in which the prospective citation must appear. It determines the function of this citation and therefore provides information that can be exploited to increase the relevance of the suggested citations.

As it is common practice, we evaluate our performance at this task by trying to recover the original citations found in papers that have already been published.

Perhaps the seminal piece of work in this area is He et al.'s [6] work, where they built an experimental citation recommendation system using the documents indexed by the CiteSeerX search engine as a test collection (over 450,000 documents), which was deployed as a testable system [8]. Recently, all metrics on this task and dataset were improved by applying multi-layered neural networks [7]. Other techniques have been applied to this task, such as collaborative filtering [2] and translation models [5], and other aspects of it have been explored, such as document representation [3] and context extraction [16].

2.1 Incoming link contexts

In order to make contextual suggestions as useful and relevant as possible, we argue here that we need to apply a measure of understanding to the text of the draft paper. Specifically, we hypothesize that there is a consistent relation between the type of *citing sentence* and the type of *cited sentence*.

In this paper, we specifically target *incoming link contexts*, also known as "anchor text" in the information retrieval literature, which is text that occurs in the vicinity of a citation to a document. Incoming link contexts (henceforth ILCs) have previously been used to generate vector-space representations of documents for the purpose of context-based citation recommendation. The idea is intuitive: a citation to a paper is accompanied by text that often summarizes a key point in the cited paper, or its contribution to the field. It has been found experimentally that there is useful information in these ILCs that is not found in the cited paper itself [17], and using them exclusively to generate a document's representation has proven superior to using the contents of the actual document [3]. Typically these contexts are treated as a single bag-of-words, often simply concatenated.

We propose a different approach here, where we separate the text in these contexts according to the type of sentence. All sentences of a same type from all ILCs to a same document are then indexed into the same field in a document in our index, allowing us to query by type of sentence in which the keywords appeared. Figure 1 illustrates our approach: the class of citing sentence is the query type, and for each query type we learn a set of weights to apply to finding the extracted keywords in different types of cited sentences in ILCs.

Our approach is to apply existing rhetorical annotation schemes to classify sentences in citing documents and use this segmentation of the anchor text to a citation to increase the relevance of recommendations.

For the task of recommending a citation for a given span of text, the ideal resource for classifying these spans would

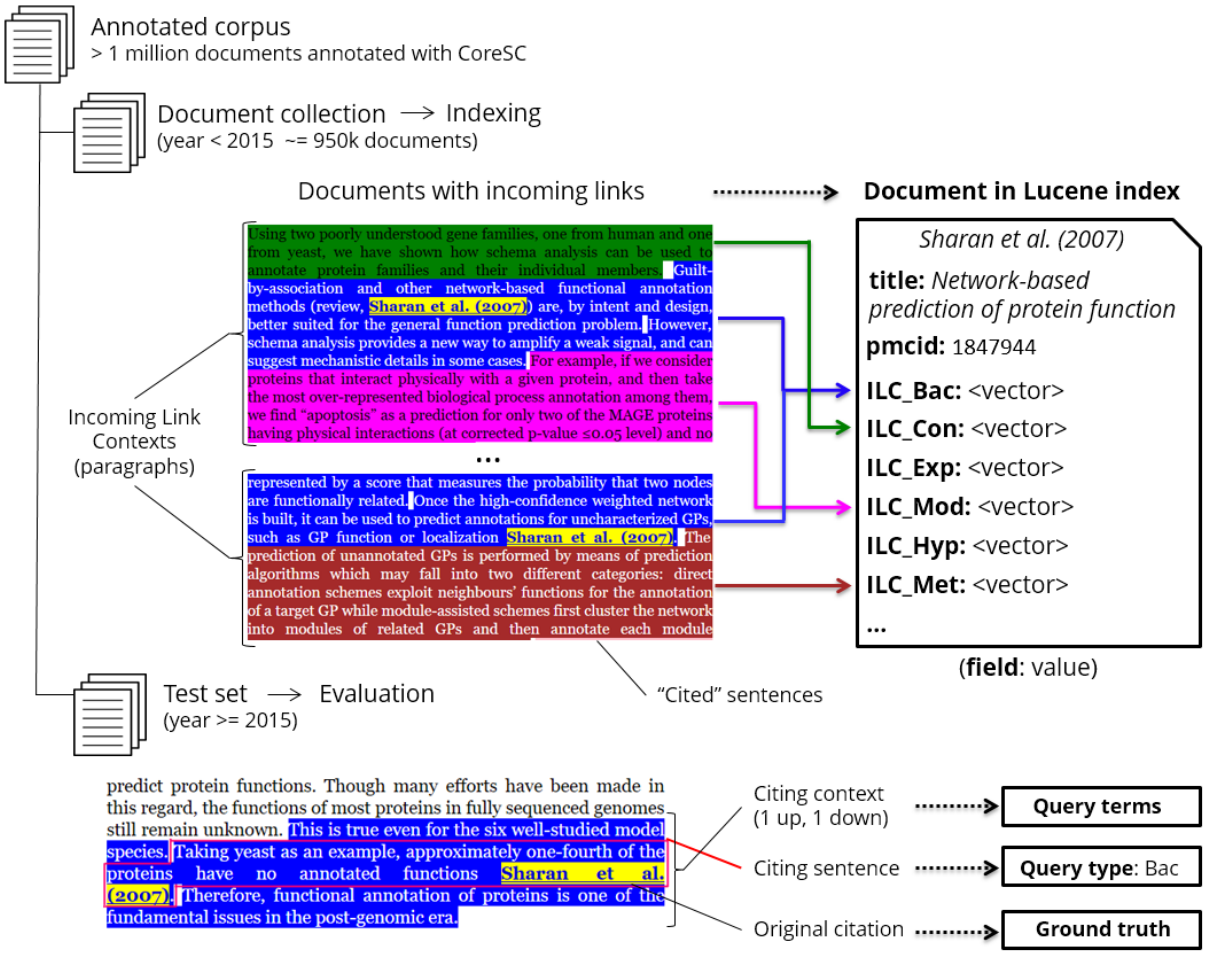

Figure 2: Indexing and query generation for evaluation using the same corpus. We use a cut-off year of publication to create our *document collection* and our *test set*. Each document in the collection is indexed containing only text from Incoming Link Contexts (ILCs) citing it from other documents in the document collection. Text from all sentences of the same CoreSC class from all ILCs to this document are indexed into a single Lucene *field*. Citations to this document from the test set are then used to generate the *queries* to evaluate on, where the *keywords* are extracted from the *citing context* (1 sentence up, 1 down, including the *citing sentence*) and the *query type* is the class of the citing sentence.

deal with the *function* of a citation within its argumentative context. While specific schemes for classifying the function of a citation have been developed (e.g. [20]), we are not aware of a scheme particularly tailored to our domain of biomedical science. Instead, we employ the CoresSC class of a citing sentence as a proxy for the function of all citations found inside it, which we have previously shown is a reasonable approach [4]. CoreSC takes the sentence as the minimum unit of annotation, continuing the standard approach to date, which we maintain in this work.

3. METHODOLOGY

We label each sentence in our corpus according with CoreSC (see Table 1), which captures its rhetorical function in the document, and we aim to find whether there is a particular link between the class of citing sentence and the class of cited text, that is, the classes of sentences found in ILCs.

As illustrated in Figure 2, we apply a cut-off date to separate our corpus into a large *document collection* and a smaller *test set* from which we will extract our queries for evaluation. We index each document in our document collection into a Lucene[1] index, creating a *field* in each document for each class of CoreSC (Hypothesis, Background, Method, etc.). We collect incoming-link contexts to all the documents in our document collection, that is, the potential documents to recommend, only from the document collection, excluding documents in our test set. We extract the paragraph where the incoming citation occurs as the ILC, keeping each sentence's label. All the text in sentences of that class from all ILCs to that document will be indexed into the same field. This allows us to apply different weights to the same keywords depending on the class of sentence they originally appeared in ILCs.

Evaluation: In order to reduce purpose-specific annotation, we use the implicit judgements found in existing sci-

[1]https://lucene.apache.org/core/

Category	Description
Hypothesis	A statement not yet confirmed rather than a factual statement
Motivation	The reasons behind an investigation
Background	Generally accepted background knowledge and previous work
Goal	A target state of the investigation where intended discoveries are made
Object-New	An entity which is a product or main theme of the investigation
Method-New	Means by which authors seek to achieve a goal of the investigation
Method-Old	A method mentioned pertaining to previous work
Experiment	An experimental method
Model	A statement about a theoretical model or framework
Observation	The data/phenomena recorded in an investigation
Result	Factual statements about the outputs of an investigation
Conclusion	Statements inferred from observations & results relating to research hypothesis

Table 1: CoreSC classes and their description. CoreSC is a content-focussed rhetorical annotation scheme developed and tested in the biomedical domain [11, 10]. Note that in this work we treat Method-Old and Method-New as a single category.

entific publications as our ground truth. That is, we substitute all citations in the text of each paper in our test set with *citation placeholders* and make it our task to match each placeholder with the correct reference that was originally cited. We only consider *resolvable citations*, that is, citations to references that point to a paper that is in our collection, which means we have access to its metadata and full machine-readable contents.

The task then becomes, for each citation placeholder, to:

1. extract its citing context, and from it the query terms (see Figure 2), and
2. attempt to retrieve the original paper cited in the context from the whole document collection

We measure how well we did at our task by how far down the list of ranked retrieval results we find the original paper cited. We use two metrics to measure accuracy: Normalized Discounted Cumulative Gain (NDCG), a smooth discounting scheme over ranks, and top-1 accuracy, which is just the number of times the original paper was retrieved in the first position.

Query extraction: For evaluation, the class of citing sentence becomes the *query type*, and for each type we apply a different set of per-field weights to each extracted term. We extract the context of the citation using a symmetric window of 3 sentences: 1 before the citation, the sentence containing the citation and 1 after. This is a frequently applied method [7] and is close to what has been assumed to be the optimal window of 2 sentences up, 2 down [13], while yielding fewer query terms and therefore allowing us more experimental freedom through faster queries.

Similarity: We use the default Lucene similarity formula for assessing the similarity between a query and a document (Figure 3).

$$\text{score}(q,d) = \text{coord}(q,d) \cdot \sum_{t \in q} \text{tf}(t \in d) \cdot \text{idf}(t)^2 \cdot \text{norm}(t,d)$$

Figure 3: Default Lucene similarity formula

In this formula, the coord term is an absolute multiplier of the number of terms in the query q found in the document d, tf is the absolute term frequency score of term t in document d, $\text{idf}(t)$ is the inverse document score and norm is a normalization factor that divides the overall score by the length of document d. Note that all these quantities are per-field, not per-document.

Technical implementation: We index the document collection using the Apache Lucene retrieval engine, specifically through the helpful interface provided by elasticsearch 2.2[2]. For each document, we create one field for each CoreSC class, and index into each field all the words from all sentences in the document that have been labelled with that class.

The *query* is formed of all the terms in the citation's context that are not in a short list of stopwords. Lucene queries take the basic form *field:term*, where each combination of *field* and *term* form a unique term in the query. We want to match the set of extracted terms to all fields in the document, as each field represents one class of CoreSC.

The default Lucene similarity formula (Figure 2) gives a boost to a term matching across multiple fields, which in our case would introduce spurious results. To avoid this, we employ DisjunctionMax queries, where only the top scoring result is evaluated out of a number of them. Having one query term for each of the classes of CoreSC for each distinct token (e.g. *Bac:"method"*, *Goa:"method"*, *Hyp:"method"*, etc.), only the one with the highest score will be evaluated as a match.

Weight training: Testing all possible weight combinations is infeasible due to the combinatorial explosion, so we adopt the greedy heuristic of trying to maximise the objective function at each step.

Our weight training algorithm can be summarized as "hill climbing with restarts". For each fold, and for each citation type, we aim to find the best combination of weights to set on sentence classes that will maximise our metric, in this case the NDCG score that we compute by trying to recover the original citation. We keep the queries the same in structure and term content and we only change the weights applied to each field in a document to recommend. Each field, as explained above, contains only the terms from the sentences in the document of one CoreSC class.

The weights are initialized at 1 and they move by -1, 6, and -2 in sequence, going through a minimum of 3 iterations. Each time a weight movement is applied, it is only kept if the score increases, otherwise the previous weight value is restored.

[2]https://elastic.co/

Query type	Num. queries	Fold	Weights Bac	Con	Exp	Goa	Hyp	Met	Mod	Mot	Obj	Obs	Res	Scores NDCG*	Accuracy*	% imp.
Bac	1000	1	2	0	0	0	0	1	0	0	0	0	1	0.290	0.120	14.54
		2	2	1	0	0	0	1	1	0	0	0	1	0.215	0.080	36.22
		3	2	0	0	0	0	1	0	0	0	0	0	0.270	0.100	31.68
		4	2	0	0	0	0	1	0	0	0	0	1	0.209	0.068	21.07
Con	278	1	2	0	0	0	0	1	0	0	0	0	1	0.242	0.100	24.14
		2	2	0	0	0	0	1	0	0	0	0	0	0.249	0.100	27.74
		3	7	6	0	0	0	1	0	0	0	0	0	0.115	0.014	30.03
		4	7	0	0	0	0	0	0	0	0	0	0	0.149	0.072	7.78
Goa	49	1	7	1	0	1	0	1	0	0	1	0	0	0.128	0.000	113.27
		2	6	1	0	1	0	1	0	1	1	0	0	0.314	0.167	49.28
		3	7	1	0	1	0	1	0	0	1	0	0	0.252	0.083	11.50
		4	7	1	0	0	0	1	0	0	0	0	0	0.148	0.083	23.14
Hyp	91	1	7	1	0	0	0	1	1	0	0	0	1	0.182	0.087	201.92
		2	7	1	0	0	0	1	0	0	0	0	1	0.240	0.087	53.01
		3	7	1	0	0	0	0	1	0	0	0	1	0.209	0.087	36.77
		4	7	0	0	0	0	1	1	0	0	0	1	0.258	0.045	85.81
Met	893	1	1	1	0	0	0	1	0	0	0	1	0	0.328	0.138	2.92
		2	1	0	0	0	1	1	0	0	0	0	1	0.328	0.130	11.39
		3	1	0	0	0	0	1	0	0	0	0	1	0.377	0.139	9.35
		4	1	1	0	0	0	1	0	0	0	0	1	0.292	0.112	10.70
Obj	70	1	1	0	0	0	0	1	0	0	0	0	1	0.148	0.056	23.29
		2	1	0	0	0	0	1	0	0	0	0	1	0.302	0.167	29.32
		3	1	0	0	0	0	1	0	0	0	0	1	0.281	0.118	8.04
		4	1	0	0	0	0	1	0	0	0	0	1	0.157	0.059	40.53
Res	420	1	2	1	1	0	0	0	0	0	0	0	1	0.150	0.057	22.81
		2	2	1	0	0	0	1	0	0	0	1	1	0.219	0.086	36.52
		3	2	1	0	0	0	0	0	0	0	0	1	0.283	0.133	34.13
		4	7	0	0	0	1	1	1	0	0	0	1	0.263	0.105	3.44

Figure 4: Weight values for the query types (types of citing sentences) that improved across all folds. The weight values for the 4 folds are shown, together with test scores and improvement over the baseline. These weights apply to text indexed from sentences in ILCs to the same document and the weight cells are shaded according to their value, darker is higher. In bold, citation types that consistently improve across folds. On the right-hand side are the scores obtained through testing and the percentage increase over the baseline, in which all weights were set to 1. *NDCG and Accuracy (top-1) are averaged scores over all citations in the test set for that fold.

This simple algorithm is not guaranteed to find a globally optimal combination of parameters for the very complex function we are optimizing, but it is sufficient for our current objective. We aim to apply more robust parameter tuning techniques to learning the weights in future work.

4. EXPERIMENTS

Our corpus is formed of one million papers from the PubMed Central Open Access collection[3]. These papers are already provided in a clean, hand-authored XML format with a well-defined XML schema[4]. For our experiments we used all papers published up to and including 2014 as our document collection (∼950k documents), and selected 1000 random papers published in or after 2015 as our *test set*. We treat the documents in the test set as our "draft" documents from which to extract the citations that we aim to recover and their citation contexts. We generate the queries from these contexts and the *query type* is the CoreSC class of the citing sentence. These are to our knowledge the largest experiments of this kind ever carried out with this corpus.

We need to test whether our conditional weighting of text spans based on CoreSC classification is actually reflecting some underlying truth and is not just a random effect of the dataset. To this end, we employ 4-fold cross-validation, where we learn the weights for 3 folds and test their impact on one fold, and we report the averaged gains over each fold.

The full source code employed to run these experiments and instructions on how to replicate them are available on GitHub[5]. The automatically annotated corpus is currently available on request, and we aim to make it publicly available shortly.

5. RESULTS AND DISCUSSION

Figure 4 shows the results for the 7 classes of citing sentences for which there was consistent improvement across all 4 folds, with a matrix of the best weight values that were found for each fold. On the right-hand side are the testing scores obtained for each fold and the percentage increase over the baseline, in which all weights are set to 1. For the remaining 4 classes (Experiment, Model, Motivation, Observation) the experiments failed to find consistent improvement, with wild variation across folds.

As is to be expected, the citations are skewed in numbers towards some CoreSC classes. A majority of citations occur within sentences that were automatically labelled Back-

[3]http://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/
[4]http://jats.nlm.nih.gov/
[5]https://github.com/danieldmm/minerva

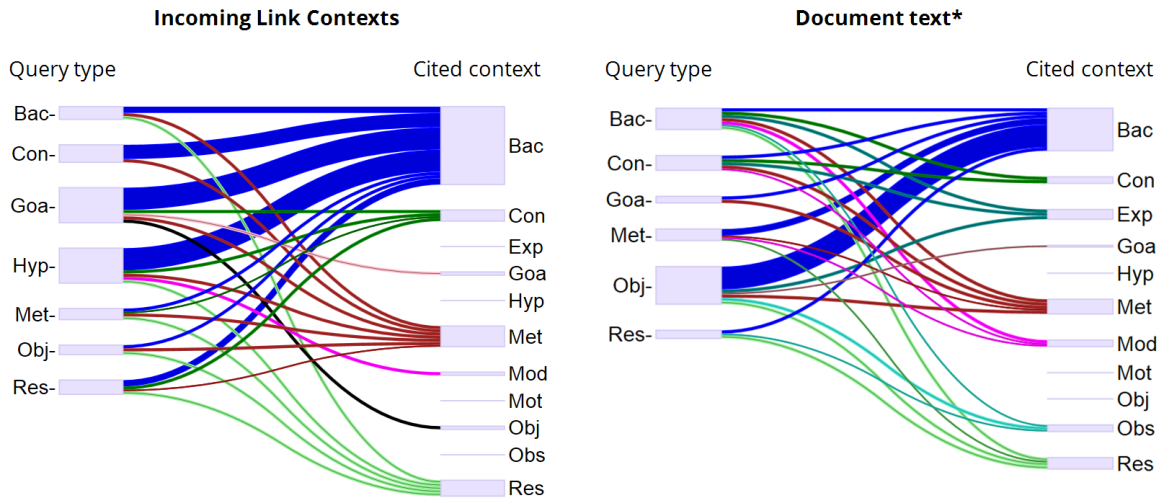

Figure 5: Citation network: links between query types and classes of cited sentences. On the left, the results presented here of CoreSC-labelled incoming link contexts. *On the right, a comparison with previous work (see [4]), where we explored the link between citing sentences and CoreSC-labelled contents of the cited document. The thickness of the lines represents the weight given to terms indexed from that class of cited sentence.

ground, Methodology and Results, no doubt due to a pattern in the layout of the content of articles. This yields many more Bac, Met and Res citations to evaluate on, and for this reason we set a hard limit to the number of citations per CoreSC to 1000 in these experiments.

A number of patterns are immediately evident from these initial results. For all query types, it seems to be almost universally useful to know that Background or Methodology sentences in a document's incoming link contexts match the query terms extracted from the citation context. The possibility exists that this is partly an effect of there being more sentences of type Background and Method in our collection.

Similarly, it seems it is better to ignore other classes of sentences in the incoming link contexts of candidate papers, specifically Experiment, Hypothesis Motivation and Observation. Also notably, Conclusion seems to be relevant only to queries of type Goal, Hypothesis and Result. Even more notably, Goal and Object seem relevant to Goal queries, and exclusively to them.

Note here that the fact that a weight combination was found where the best weight for a citing sentence class is 0 does not mean that including information from this CoreSC is *not useful* but rather that it is in fact *detrimental*, as eliminating it actually increased the average NDCG score. These are of course averaged results, and it is certain that the weights that we find are not optimal for each individual test case, only better on average.

It is important to note that our evaluation pipeline necessarily consists of many steps, and encounters issues with XML conversion, matching of citations with references, matching of references in papers to references in the collection, etc., where each step in the pipeline introduces a measure of error that we have not estimated here. The one we can offer an estimate for is that of the automatic sentence classifier. The Sapienta classifier[6] we employ here has recently been independently evaluated on a different corpus from the originally annotated corpus used to train it. It yielded 51.9% accuracy over all eleven classes, improving on the 50.4% 9-fold cross-validation accuracy over its training corpus [15].

Further to this, we judge that the consistency of correlations we find confirms that what we can see in Figure 4 is not due to random noise, but rather hints at underlying patterns in the connections between scientific articles in the corpus.

Figure 5 shows our results as a graph, with the per-class weights flowing from the class of citing sentence to the class of cited sentence. For this graph, we take a "majority vote" for the weights from Figure 4: if three folds agree and a fourth differs by a small value, we take this to be noise and use the majority value. If folds agree in two groups we average the values.

We show a side-by-side comparison of these new results with our previous results where we indexed a document's actual contents instead of the incoming link contexts to it. We had previously proposed that there is an observable link between the class of citing sentence and the class of sentence in the cited document [4]. Now we find the same evidence for a link between the class of citing sentence and the class of sentence within incoming link contexts, so inside *other* documents citing a given document.

There are both similarities and differences between the weights found for incoming link contexts and document text. Background and Method are almost as universally relevant for one as for the other, and Results equally as irrelevant for citing sentences of classes Conclusion and Goal. However, we also find that whereas sentences of type Observation found inside a document's text are useful (for Background, Object and Result), they are not when they are found inside incoming link contexts to that document.

6. CONCLUSION AND FUTURE WORK

We have presented a novel application of CoreSC discourse function classification to context-based citation recommen-

[6]http://www.sapientaproject.com

dation, an information retrieval application. We have carried out experiments on the full PubMed Central Open Access Corpus and found strong indications of correlation between different classes of sentences in the Incoming Link Contexts of documents citing a single document. We also find that these relationships are not intuitively predictable and yet consistent.

This suggests that there are gains to be reaped in a practical application of CoreSC to context-based citation recommendation. In future work we aim to evaluate this against more standard approaches, such as concatenating and indexing the anchor text and the document text together.

7. REFERENCES

[1] M. Angrosh, S. Cranefield, and N. Stanger. Context identification of sentences in research articles: Towards developing intelligent tools for the research community. *Natural Language Engineering*, 19(04):481–515, 2013.

[2] C. Caragea, A. Silvescu, P. Mitra, and C. L. Giles. Can't see the forest for the trees?: a citation recommendation system. In *Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries*, pages 111–114. ACM, 2013.

[3] D. Duma and E. Klein. Citation resolution: A method for evaluating context-based citation recommendation systems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers)*, page 358âĂŞ363, Baltimore, Maryland, USA, 2014.

[4] D. Duma, M. Liakata, A. Clare, J. Ravenscroft, and E. Klein. Applying core scientific concepts to context-based citation recommendation. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, 2016.

[5] J. He, J.-Y. Nie, Y. Lu, and W. X. Zhao. Position-aligned translation model for citation recommendation. In *String Processing and Information Retrieval*, pages 251–263. Springer, 2012.

[6] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 421–430. ACM, 2010.

[7] W. Huang, Z. Wu, C. Liang, P. Mitra, and C. L. Giles. A neural probabilistic model for context based citation recommendation. In *AAAI'15 Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[8] W. Huang, Z. Wu, P. Mitra, and C. L. Giles. Refseer: A citation recommendation system. In *Digital Libraries (JCDL), 2014 IEEE/ACM Joint Conference on*, pages 371–374. IEEE, 2014.

[9] K. Hyland. *Academic discourse: English in a global context*. Bloomsbury Publishing, 2009.

[10] M. Liakata, S. Saha, S. Dobnik, C. Batchelor, and D. Rebholz-Schuhmann. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000, 2012.

[11] M. Liakata, S. Teufel, A. Siddharthan, and C. R. Batchelor. Corpora for the conceptualisation and zoning of scientific papers. In *LREC*, 2010.

[12] P. I. Nakov, A. S. Schwartz, and M. Hearst. Citances: Citation sentences for semantic analysis of bioscience text. In *Proceedings of the SIGIRâĂŹ04 workshop on Search and Discovery in Bioinformatics*, pages 81–88, 2004.

[13] V. Qazvinian and D. R. Radev. Identifying non-explicit citing sentences for citation-based summarization. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 555–564. Association for Computational Linguistics, 2010.

[14] J. Ravenscroft, M. Liakata, and A. Clare. Partridge: An effective system for the automatic cassification of the types of academic papers. In *Research and Development in Intelligent Systems XXX*, pages 351–358. Springer, 2013.

[15] J. Ravenscroft, A. Oellrich, S. Saha, and M. Liakata. Multi-label annotation in scientific articles -the multi-label cancer risk assessment corpus. In *Proceedings of the 10th edition of the Language Resources and Evaluation Conference*, 2016.

[16] A. Ritchie. Citation context analysis for information retrieval. Technical report, University of Cambridge Computer Laboratory, 2009.

[17] A. Ritchie, S. Teufel, and S. Robertson. Creating a test collection for citation-based ir experiments. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 391–398. Association for Computational Linguistics, 2006.

[18] U. Schäfer and U. Kasterka. Scientific authoring support: A tool to navigate in typed citation graphs. In *Proceedings of the NAACL HLT 2010 workshop on computational linguistics and writing: Writing processes and authoring aids*, pages 7–14. Association for Computational Linguistics, 2010.

[19] S. Teufel. *Argumentative zoning: Information extraction from scientific text*. PhD thesis, Citeseer, 2000.

[20] S. Teufel, A. Siddharthan, and D. Tidhar. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 103–110. Association for Computational Linguistics, 2006.