Identifying Meaningful Citations

Marco Valenzuela
University of Arizona
Tucson, AZ, USA
marcov@email.arizona.edu

Vu Ha and **Oren Etzioni**
Allen Institute for Artificial Intelligence
Seattle, WA, USA
{vuh,orene}@allenai.org

Abstract

We introduce the novel task of identifying important citations in scholarly literature, i.e., citations that indicate that the cited work is used or extended in the new effort. We believe this task is a crucial component in algorithms that detect and follow research topics and in methods that measure the quality of publications. We model this task as a supervised classification problem at two levels of detail: a coarse one with classes (important vs. non-important), and a more detailed one with four importance classes. We annotate a dataset of approximately 450 citations with this information, and release it publicly. We propose a supervised classification approach that addresses this task with a battery of features that range from citation counts to where the citation appears in the body of the paper, and show that, our approach achieves a precision of 65% for a recall of 90%.

Introduction

Tracking citations is an important component of analyzing scholarly big data. Citations provide a quantitative way to measure the quality of published works, to detect emerging research topics, and to follow evolving ones.

In this work we argue that not all citations are equal. While some indeed indicate that the cited work is used or, more importantly, extended in the new publication, some are less important, e.g., they discuss the cited work in the context of related work that does not directly impact the new effort. To illustrate this point, Table 1 lists several citations in increasing order of importance. We further argue that because current citation tracking algorithms do not distinguish between important vs. incidental citations, all of the above applications (e.g., measuring the quality of a publication, or tracking research topics) are negatively affected.

To our knowledge, this work is among the first to tackle the problem of identifying important citations. The contributions of our work are the following:

1. We introduce the novel task of identifying important citations, defined as a classification task with either two classes (important vs. non-important citation) or four classes (following the examples in Table 1).

Citation Type	Citation Text
incidental: related work	Discriminative models have recently been proved to be more effective than generative models in some NLP tasks, e.g., parsing (Collins 2000), POS tagging (**Collins 2002**) and LM for speech recognition (Roark et al. 2004).
incidental: comparison	Online baselines include Top-1 Perceptron (**Collins, 2002**), Top-1 Passive-Aggressive (PA), and k-best PA (Crammer & Singer, 2003; McDonald et al., 2004).
important: using the work	Here, we follow the definition of Collins perceptron (**Collins, 2002**). The part-of-speech tagger is our reimplementation of the work in (**Collins, 2002**).
important: extending the work	We describe a new sequence alignment model based on the averaged perceptron (**Collins, 2002**), which shares with the above... Our learning method is an extension of Collins's perceptron-based method for sequence labeling (**Collins, 2002**).

Table 1: Citation examples for *(Collins, 2002)*, listed in increasing order of importance.

2. We annotate a dataset of approximately 450 citations with citation importance information. The dataset is publicly available in the hope that it will foster further research on this topic.

3. We propose a supervised classification approach that separates important from incidental citations using a battery of features, ranging from citation counts to where the citation appears in the body of the paper. Our approach models both direct citations, i.e., citations that follow established proceeding formats, and indirect citations, i.e., which use a description of the algorithm instead (e.g., "Stanford parser"). Our method performs well, obtaining a precision of 65% for a recall of 90%.

Data

To address this task, we used a collection of 20,527 papers from the ACL anthology[1], together with their citation graph

[1] http://www.aclweb.org/anthology/

Citation Type	Fine-grained Label	Coarse Label
Related work	0	Incidental
Comparison	1	Incidental
Using the work	2	Important
Extending the work	3	Important

Table 2: Citation annotation labels.

and metadata generated by (Elkiss et al. 2008). There were 106,509 citations among them. We annotated 465 of these citations, represented as tuples of *(cited paper, citing paper)*, with ordinal labels ranging from 0 to 3, in increasing order of importance. The four classes follow the examples in Table 1. To obtain a coarse, binary label set, we also collapsed these fine-grained labels, such that 0 and 1 indicate incidental citations, and 2 and 3 indicate important citations. Table 2 summarizes both the fine-grained and the coarse label sets.

The citations were annotated by one expert, but we verified inter-annotator agreement between two experts for a subset of the dataset. For the set of four fine-grained labels, the annotators agreed on 83.6% of the citations; for the coarse label set, the inter-annotator agreement was 93.9%. These results indicates that this task is relatively easy for domain experts.

Crucially, we found that in this dataset only 14.6% of the annotated citations are considered important, i.e., they are labeled 2 (using the cited work) or 3 (extending the work). This further demonstrates that the identification of important citations is an important task, since most citations are actually incidental.

The dataset of papers behind these citations was preprocessed as follows:

- To extract the text from the PDF files we used Poppler's `pdftotext`[2].
- The text was normalized by removing diacritics with an in-house script.
- For sentence splitting, tokenization and POS tagging we used Factorie (McCallum, Schultz, and Singh 2009).
- For shallow parsing, or chunking, we used OpenNLP[3].
- To identify the section in which the citation occurs, we used ParsCit to segment the paper into sections (Luong, Nguyen, and Kan 2010). ParsCit provides normalized section names, which we use instead of the section titles. For example, both "State of the Art", and "Previous Work" are normalized to `related_work`.

Approach

Our modeling of citation importance is driven by three key observations:

1. The more citations a paper receives in the body of the citing work, the more important the citation is likely to be.
2. It matters where the citation appears. For example, a citation in the Related Work section is likely to indicate an incidental citation. On the other hand, a citation in the Methods section indicates that the cited work is used or extended in the citing paper, which signals importance.
3. Citations appear in many forms. Some are *direct*, i.e., the citation follows an established proceedings format, or *indirect*, where the work is cited by mentioning the name of an author, typically the first author, the name of the cited algorithm, of a description of the algorithm. Table 3 shows examples of indirect citations. Thus, in order to reliably implement the first two observations, one has to first identify both direct and indirect citations.

Citation Type	Citation Text
Algorithm name	Figure 3: **Stanford Parser** output example.
Algorithm name	We implement a part-of-speech tagger with **averaged perceptron**.
Algorithm name	We implemented **the MXPOST tagger** and integrated it with our algorithm.
Author + Algorithm name	However, the application of **the Yarowsky algorithm** to NER involves several domain-specific choices as will become evident below.
Author	The behaviour is slightly different here, with **Charniak** obtaining better results than **Bikel** in most cases.

Table 3: Indirect citations by name of first author or name/description of the cited algorithm.

Identifying Direct and Indirect Citations

We identified direct citations using rules that follow the citation format of the ACL proceedings, and matched them to unique paper identifiers in our corpus. A regular expression was generated using the paper metadata. This regular expression was designed to match citations that follow the format of the ACL proceedings and some variations that occurred in our corpus. For example, the required syntax for a citation to a paper with three authors is to write the last name of the first author followed by the phrase "et al." but we found that many papers in our corpus mention all the authors last names.

To identify indirect citations, we implemented two heuristics: one focusing on author names, and one addressing names or descriptions of the cited algorithms. We extracted indirect citations by author name by first finding all citations to any paper in the citing paper's text and then matching the last name of the first author of the cited paper of interest outside any of the direct citations.

Automatically identifying algorithm name or descriptions is less trivial. For this, we implement a two step algorithm:

1. For any given cited paper, we first find the papers that cite it in the entire corpus of 20K+ papers, and we extract the corresponding citations. Then we extract: (a) the noun phrase directly before the citation, or (b) the noun phrase following the citation and a verb. For this step, we used the Knowitall Taggers tool, a pattern-matching tool that functions over tokens and incorporates part-of-speech and

[2] `http://poppler.freedesktop.org`

[3] `http://opennlp.apache.org`

Cited paper id	Verb	Noun phrase
P05-1044	proposed	a new objective function
J96-1002	presented	a Maximum Entropy Approach
A00-1031	reports	96.7% overall accuracy
W06-1643	used	skip-chain Conditional Random Fields
P08-1108	combined	MSTParser and MaltParser
W02-1001	extended	the perceptron algorithm
P02-1018	reports	93% precision and 83% recall

Table 4: Examples of noun phrases following citations. The paper ids are from the ACL ontology; to retrieve the paper content append the id at the end of this URL: `http://www.aclweb.org/anthology/`.

shallow syntactic information[4]. Table 4 shows examples of citations followed by a verb and a noun phrase. Most of these noun phrases are informative, but some do not describe the cited approach, e.g., focusing instead on its "accuracy" and "recall". We address these errors below with a robust heuristic inspired from information retrieval.

2. In the second step, we collect the unigrams and bigrams found in the noun phrases related to each cited paper and identify the most important ones by selecting the ones with a *tf-idf* score (Manning, Raghavan, and Schütze 2008) above some threshold (arbitrarily set to 200). Table 5 shows examples of the extracted names and descriptions, which illustrates that this filtering manages to remove most of noise introduced in the previous step. A remaining limitation of our approach is that we currently use unigrams and bigrams, which might not to be sufficient to capture longer descriptions. We analyze this issue later in the paper.

Features

Using both direct and indirect citations, we extract the following features from each citation tuple:

- **(F1) Total number of direct citations:** This feature counts the total number of citations to the cited paper.
- **(F2) Number of direct citations per section:** Similar to the above feature, but counts are qualified by the section in which they appear. For this feature we used the normalized section titles produced by ParsCit. For example, if a paper has five citations, with two appearing in the Related Work section and three in Methods, we generate two features: `DirectCountsRelatedWork` with a value of 2 and `DirectCountsMethods` with a value of 3.
- **(F3) Total number of indirect citations** and **number of indirect citations per section**: Similar to the previous two features but focusing on indirect features. Since the description n-grams may be redundant (i.e., we may find multiple, slightly different descriptions of the same work) we count them differently than direct citations: instead of counting occurrences, we count the number of sentences in which at least one potential description appears.
- **(F4) Author overlap** (Boolean): This feature is set to true if the citing and the cited works share at least one common author. The intuition behind this feature is that shared authors indicate that the new work is likely to be an extension of the cited paper.
- **(F5) Is considered helpful** (Boolean): This feature is set to true if a sentence in which a citation occurs contains phrases such as "we follow" or "we used", which are hints that the author of the citing work considers the cited paper to be important.
- **(F6) Citation appears in table or caption** (Boolean): Set to true if at least a citation appears in a table or a caption of a figure or table. This is an indicator that the author of the citing work is comparing her results to the cited paper.
- **(F7) 1 / number of references:** This feature computes the inverse of the length of the citing paper's reference list, which hints to the value of receiving one citation, e.g., if it is one citation from a total of two references, this citation is clearly important.
- **(F8) Number of paper citations / all citations:** Similarly, this feature computes the number of direct citations instances for the cited paper over all the direct citation instances in the citing work.
- **(F9) Similarity between abstracts:** This feature computes the similarity between the cited and citing paper's abstracts using the cosine similarity of the *tf-idf* scores. The intuition behind this feature is that the closer the abstracts, the more likely the new work extends the cited paper.
- **(F10) PageRank:** This feature computes the PageRank score (Page et al. 1999) of the cited paper, as a measure of the cited work's importance.
- **(F11) Number of total citing papers after transitive closure:** This feature records the number of citing papers after the transitive closure, e.g., papers that cite the cited work, papers that cite those papers, etc.
- **(F12) Field of the cited paper:** This feature stores the particular computer science subfield to which the cited paper belongs. This is work in progress: we currently developed a classifier that identifies if a paper describes a software system or not. This classifier was developed as part of a scientific literature search engine and is based on bag of words technique matching system names with citation contexts.

For learning, we used classifiers implemented in the `scikit learn` toolkit[5], in particular support vector machines (SVM) and random forests. We normalized all numeric features by centering on the mean and scaling to unit variance.

Experiments

Identifying important citations

For the main experiments in this section, i.e., identifying important citations, we used a leave-one-out cross-validation setup, i.e., we repeatedly evaluated the performance of our

[4] `https://github.com/knowitall/taggers`

[5] `http://scikit-learn.org/`

Citation	Paper title	Nickname	TF-IDF score
P03-1054	Accurate Unlexicalized Parsing	stanford parser	1236.89
W04-3252	TextRank: Bringing Order into Texts	textrank	319.33
N07-1051	Improved Inference for Unlexicalized Parsing	berkeley parser	506.45
A00-1031	TnT – A Statistical Part-of-Speech Tagger	tnt	1041.02
W96-0213	A Maximum Entropy Model for Part-Of-Speech Tagging	mxpost	377.71
W02-1001	Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms	structured perceptron	356.06
W02-1001	Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms	averaged perceptron	373.52

Table 5: Identified algorithm names/descriptions.

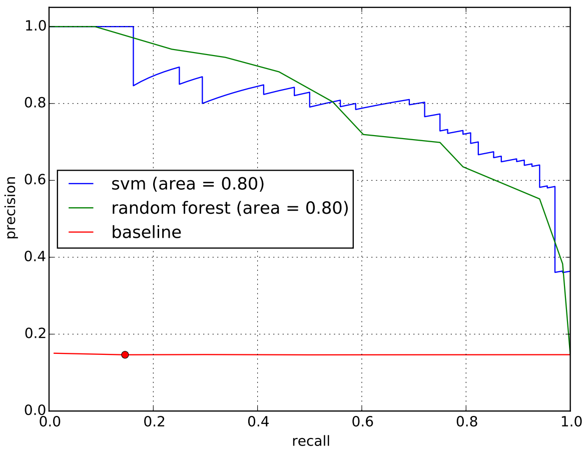


Figure 1: Precision-recall curve for our baseline and two classifiers: SVM with a RBF kernel and random forests.

models on a different citation, by training on all remaining ones. For this experiment, we used only the binary coarse labels, i.e., important vs. incidental, and employed the standard precision, recall, and F1 scores as evaluation measures, considering the important citations as the positive class.

Figure 1 shows the results of our model trained with two classifiers: SVM with a RBF kernel, and random forests. To obtain the P/R curve, we used various thresholds on the classifier confidence. Both classifiers are compared against a baseline that randomly assigns the “important” label using a probability p, which varies from 0 to 1. The red dot in the figure corresponds to a value of p equal to the the prior distribution of the “important” label in the entire corpus (i.e., 14.6%).

The results in the figure show that our proposed model considerably outperforms the baseline: for example, for a recall of 0.9 our SVM model has a precision of approximately 0.65, whereas the baseline’s precision at the same recall point is under 0.2. This is an important result, which shows that under a high-recall requirement (which is a common scenario for a real-life system – see the Discussion section) our system has a reasonable precision. Overall, both classifiers have an area under the curve of 0.80. We consider this a very encouraging result for our relatively simple model.

Figure 2 shows the learning curve of the SVM classifier. For this experiment, we used a simpler, three-fold cross validation. For each testing fold, we randomly selected subsets of the training data for each point in the curve. To avoid potential biases in the random subset selection, we repeated the experiments five times and averaged the results. This experiment indicates that the classifier learns relatively quickly, achieving near optimal performance with half of the training data available. This suggests that, even though our corpus is relatively small, its size is not a drastic constraint on performance.

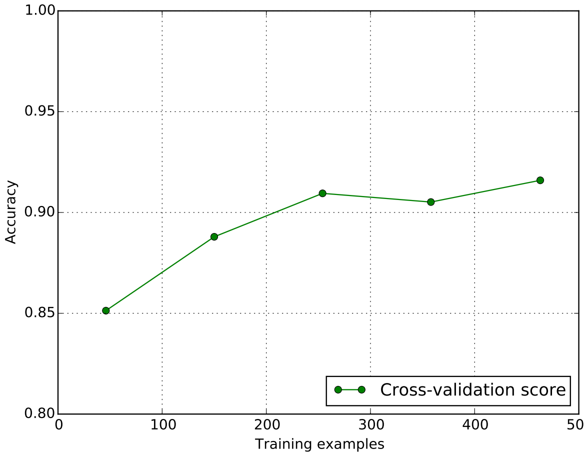


Figure 2: Learning curve for the SVM classifier with RBF kernel.

Feature analysis

To understand the contribution of each of the features proposed in the previous section, we performed a *post-hoc* analysis, where we evaluated variants of our model containing a *single* feature group at a time. Because we are ultimately interested in a high-recall configuration of the classifier, where no important citations are missed (see the Discussion section), for this analysis we enforced a high recall of 0.9 for all configurations. The results are listed in Table 6.

The table highlights that all individual features perform better the random baseline. Recall that the baseline had both precision and recall under 0.2, whereas most of our features have a precision of over 0.2 for a recall of 0.9. The fact that all features contribute to the overall performance is highlighted by the performance of the system that uses all features (last row in the table), which is nearly double that of the best performing individual feature.

However, the individual feature contributions vary widely.

Features	Precision
Only: direct citations per section (F2)	0.37
Only: direct citations (F1)	0.30
Only: author overlap (F4)	0.22
Only: is useful (F5)	0.22
Only: direct citations / all citations (F8)	0.22
Only: research field (F12)	0.22
Only: in figure/table (F6)	0.20
Only: indirect citations: author names (F3)	0.19
Only: indirect citations: descriptions (F3)	0.17
Only: inverse number of references (F7)	0.17
Only: PageRank (F10)	0.17
Only: total citing papers (F11)	0.16
Only: abstract similarity (F9)	0.14
All features	0.62

Table 6: Performance of the system when using individual feature groups, for a recall of 0.90. The feature groups are listed in descending order of their contribution.

The best performing features are the direct citations (both globally and per section) (F2, F1), followed by author overlap between citing and cited papers (F4), and textual hints that the cited work is considered useful by the authors of the citing paper (F5). The least performing features are: our PageRank score (F10) (perhaps due the small size of our paper dataset), the total number of citing papers after transitive closure (F11) (which suggests that influence dissipates beyond the immediate citations), and similarity of abstracts (F9) (suggesting that researchers working on similar topics are not necessarily influencing each other).

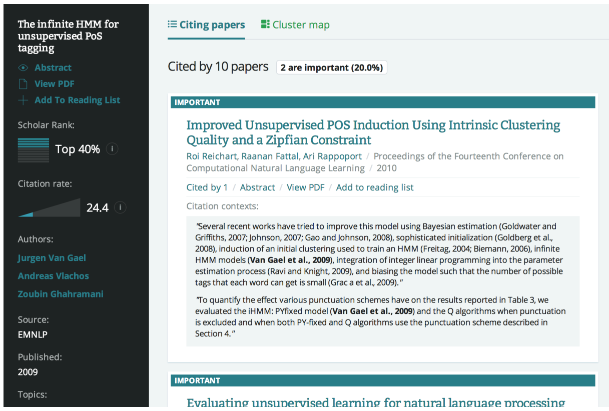
Evaluating paper descriptions

Although the paper descriptions extracted by our algorithm are informative (see Table 5), the analysis in the previous sub-section indicates that indirect citations, which use these descriptions, contribute minimally to the overall performance. To better understand this issue, we performed a direct evaluation of the descriptions that our algorithm extracts.

In this work we focus solely on the precision of the set of extracted descriptions, which are responsible for the indirect citations.[6] For this purpose, we selected a subset of 50 papers from the larger corpus of +20K papers. 12 of these papers appear as cited papers in our smaller, annotated citation corpus. For these 50 papers, we collected all the descriptions automatically extracted by our approach. The descriptions in this set total 119. A domain expert analyzed these descriptions, and produced two precision scores: a *lenient* score, which considers a n-gram description as correct if it is part of a correct description, and a *strict* score, which considers a description as correct only if it forms a complete, non-ambiguous description. For example, for the paper "Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling"[7], the expert considered the description "entity recognizer" correct under the lenient score but incorrect under the strict one, whereas the description "stanford ner" is marked as correct under both scores.

The results of this evaluation were a lenient precision score of $115/119 = 96.6\%$, and a strict precision of $46/119 = 38.7\%$. This analysis indicates that the n-grams extracted are almost always relevant (hence the high lenient score), but seldom complete (hence the low strict score). For example, for the above paper, our algorithm extracts the following descriptions: "stanford named", "named", "recognizer", "named entity", "entity recognizer", "stanford", and "stanford ner". While these are clearly relevant for this paper, the incomplete descriptions, e.g., "named entity", may have two undesired effects: (a) they are likely to match in the context of another paper, yielding incorrect indirect citations, and (b) they may cause spurious citations, when multiple incomplete descriptions that form a complete one (e.g., "stanford named" and "entity recognizer") match in the same sentence. In this work, we mitigate the latter issue by counting sentences rather than individual indirect citations. In future work, we will explore more complex solutions, such as n-gram tiling (Dumais et al. 2002), which combine multiple incomplete n-grams to form a complete description.

Discussion



The infinite HMM for unsupervised PoS tagging

Scholar Rank: Top 40%

Citation rate: 24.4

Authors:
Jurgen Van Gael
Andreas Vlachos
Zoubin Ghahramani

Source:
EMNLP

Published:
2009

Topics:

Cited by 10 papers 2 are important (20.0%)

Improved Unsupervised POS Induction Using Intrinsic Clustering Quality and a Zipfian Constraint

Roi Reichart, Raanan Fattal, Ari Rappoport / Proceedings of the Fourteenth Conference on Computational Natural Language Learning / 2010

Citation contexts:

"Several recent works have tried to improve this model using Bayesian estimation (Goldwater and Griffiths, 2007; Johnson, 2007; Gao and Johnson, 2008), sophisticated initialization (Goldberg et al., 2008), induction of an initial clustering used to train an HMM (Freitag, 2004; Biemann, 2006), infinite HMM models (**Van Gael et al., 2009**), integration of integer linear programming into the parameter estimation process (Ravi and Knight, 2009), and biasing the model such that the number of possible tags that each word can get is small (Grac a et al., 2009)."

"To quantify the effect various punctuation schemes have on the results reported in Table 3, we evaluated the iHMM: PYfixed model (**Van Gael et al., 2009**) and the Q algorithms when punctuation is excluded and when both PY-fixed and Q algorithms use the punctuation scheme described in Section 4."

Evaluating unsupervised learning for natural language processing

Figure 3: Screenshot of the scientific literature search engine that uses this work. For each cited paper, the top right block lists the important citations out of the total citations found in the indexed corpus.

One of the strengths of this work is its immediate applicability to a real-world problem. We have incorporated our citation classifier into a scientific literature search engine, such that users can immediately identify the most important followup work for a given cited paper. Figure 3 shows a screenshot of this search engine using our work. The example highlights that of the ten papers that cite the paper "The infinite HMM for unsupervised PoS tagging" only two are considered important and are shown first. To avoid missing important citations, this system uses the high-recall configuration of our system (R 0.90, P 0.65).

This work is however far from complete. In future work we will improve the extraction of the paper description and, correspondingly, of the indirect citations, by implementing

[6] Furthermore, a recall-based evaluation in this context is hard: it is not trivial to extract all the possible descriptions of a paper in the literature.

[7] https://www.aclweb.org/anthology/P05-1045

tiling algorithms that merge incomplete descriptions. We will continue to improve the features used to represent a citation. For example, we will explore different ways of normalizing citation counts, e.g., by dividing by the total number of references and/or citations in the citing paper. Also, we will evaluate new features like the rhetorical function of citations (Teufel, Siddharthan, and Tidhar 2006).

Related Work

There has been considerable effort in the past decade on citation indexing systems (Giles, Bollacker, and Lawrence 1998; Lawrence, Giles, and Bollacker 1999; Councill, Lee, and Giles 2006) and on algorithms that analyze these citation graphs to, e.g., understand the flow of research topics in the literature, model the influence of specific papers in their field, or recommend citations for a given topic; see, inter alia, (Dietz, Bickel, and Scheffer 2007; Gruber, Rosen-Zvi, and Weiss. 2008; Nallapati et al. 2008; Daume III 2009; Sun et al. 2009; Wong et al. 2009; Nallapati, McFarland, and Manning 2011). However, by and large, these works assume that all citations are important, which we dispute in our work. We argue that by identifying the citations that are truly important, we will arrive at a better understanding of published research, which will lead to novel or more accurate applications of scholarly big data.

Our work is closest to (Zhu et al. 2013), which focuses on identifying key references for a given paper. Zhu et al. create a dataset of citations, labeled according to their influence by the authors of the citing papers, and train a supervised classifier with four features to predict academic influence. Our work is different in that our dataset of citations is annotated by (unbiased) domain experts and we explore a much larger feature set (twelve vs. four).

Conclusions

To our knowledge, this paper is among the first to tackle the important task of identifying important citations, which, we believe, will ultimately improve many applications that focus on tracking scholarly citations, such as detecting and following research trends, or quantitatively measuring the quality and impact of publications.

In addition to introducing and formalizing this task, our contributions include a novel dataset of 465 citation tuples, which is publicly available[8]. We also describe a supervised classification approach for identifying meaningful citations, which uses a battery of features ranging from citation counts to where the citation appears in the body of the paper. Using the previously described dataset, we show that our approach performs well, obtaining a precision of 0.65 for a high recall of 0.9.

References

Councill, H. L. I.; Lee, W.-C.; and Giles, C. L. 2006. Citeserx: an architecture and web service design for an academic document search engine. In *International Conference on World Wide Web*.

Daume III, H. 2009. Markov random topic fields. In *ACL-IJCNLP*.

Dietz, L.; Bickel, S.; and Scheffer, T. 2007. Unsupervised prediction of citation influences. In *International Conference on Machine Learning*.

Dumais, S.; Banko, M.; Brill, E.; Lin, J.; and Ng, A. 2002. Web question answering: Is more always better? In *SIGIR*.

Elkiss, A.; Shen, S.; Fader, A.; Erkan, G.; States, D.; and Radev, D. 2008. Blind men and elephants: What do citation summaries tell us about a research article? *J. Am. Soc. Inf. Sci. Technol.* 59(1):51–62.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. Citeseer: an automatic citation indexing system. In *ACM conference on Digital libraries*.

Gruber, A.; Rosen-Zvi, M.; and Weiss., Y. 2008. Latent topic models for hypertext. In *Uncertainty in Artificial Intelligence*.

Lawrence, S.; Giles, C. L.; and Bollacker, K. D. 1999. Digital libraries and autonomous citation indexing. *Computer* 32(6):67– 71.

Luong, M.-T.; Nguyen, T. D.; and Kan, M.-Y. 2010. Logical structure recovery in scholarly articles with rich document features. *International Journal of Digital Library Systems (IJDLS)* 1(4):1–23.

Manning, C. D.; Raghavan, P.; and Schütze, H. 2008. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.

McCallum, A.; Schultz, K.; and Singh, S. 2009. FACTORIE: Probabilistic programming via imperatively defined factor graphs. In *Neural Information Processing Systems*.

Nallapati, R. M.; Ahmed, A.; Xing, E. P.; and Cohen, W. W. 2008. Joint latent topic models for text and citations. In *KDD*.

Nallapati, R.; McFarland, D.; and Manning, C. 2011. Unsupervised learning of topic specific influences of hyperlinked documents. In *Artificial Intelligence and Statistics*.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Sun, C.; Gao, B.; Cao, Z.; and Li, H. 2009. HTM: a topic model for hypertexts. In *Empirical Methods in Natural Language Processing*.

Teufel, S.; Siddharthan, A.; and Tidhar, D. 2006. Automatic classification of citation function. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*.

Wong, C.; Thiesson, B.; Meek, C.; and Blei, D. 2009. Markov topic models. In *Artificial Intelligence and Statistics*.

Zhu, X.; Turney, P.; Lemire, D.; and Vellino, A. 2013. Measuring academic influence: Not all citations are equal. *submitted to Journal of the Association for Information Science and Technology (JASIST)*.

[8] http://allenai.org/data.html