Honours Year Project Report

Citation Provenance

By

Heng Low Wee
(U096901R)

Department of Computer Science

School of Computing

National University of Singapore

2011/12

Honours Year Project Report

Citation Provenance

By

Heng Low Wee
(U096901R)

Department of Computer Science

School of Computing

National University of Singapore

2011/12

Project No: H079820
Advisor: A/P Min-Yen Kan
Deliverables:
Report: 1 Volume
Source Code: 1 DVD

Abstract

Citations in research paper acknowledges previous work and gives the provenance to key ideas in the cited paper. However, it is difficult for a reader to locate the cited information that justifies a citation without first investing time to read through the cited paper. We investigate *Citation Provenance*, a new task in citation analysis, which means to discover the origin of the information embodied by a citation. We first describe the challenges in collecting annotations for our training set, and present a two-tier approach in tackling this problem. We adopt features previously used in Citation Classification and Information Retrieval tasks, and with them, we differentiate citations that refer to the whole paper in general (*general*) versus ones that cite specific claims, evidence or parts of the paper (*specific*). Given that a citation is *specific*, our second tier classifier localizes the cited information in the cited paper. Our first tier (*GvS*) obtained an accuracy of 0.786 in cross-validation evaluation and our second tier (*LocateProv*) showed an improvement over our baseline.

Subject Descriptors:
Information Systems[1]
- Information systems applications
Digital libraries and archives
- Information retrieval
Information extraction

Keywords:
citation analysis, citation provenance, source of citation, citation classification

Implementation Software and Hardware:

Software: Python, NLTK[2], scikit-learn[3]
Hardware: MacBook Pro, Intel Core 2 Duo 2.4GHz, 4GB Memory.

[1]Based on The 2012 ACM Computing Classification System
[2]http://nltk.org/
[3]http://scikit-learn.org/

Acknowledgement

I would like to express my gratitude to all the volunteer participants from the NUS WING group for participating in my pilot annotation tests. I thank them for testing my annotation scheme, and appreciate the feedback that improved my project.

Congrats to the `CodeForScience` team from WING NUS, consisting of Jin Zhao, Tao Chen, Eric Yulianto Ang and myself for having emerged the winners for the competition with our prototype application CitWeb that integrated our projects – Citation Classification and Citation Provenance.

Million thanks to Jin Zhao, Tao Chen, and especially my supervisor to this project, A/P Min Yen Kan for providing their guidance during the duration of the project.

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

Citing previously published scientific papers is an important practice among researchers. It gives credit and acknowledgement to original ideas and to researchers who did significant work in enabling the current research. More importantly, it upholds intellectual property. A reader of such research papers often encounters these citations made by the authors in various sentences throughout the paper. When a reader wishes to gain a better understanding of the current context, it is necessary to follow these citations and read the cited papers to understand the basis for the current work. Often, when reading the claims of a sentence supported by a citation, readers wish to know where in the cited paper the information comes from.

However, as frequent readers might find, most citations are only *mentions*. These citations are what we term *general* citations. Other citations refer specifically to particular claims, parts or sections of a paper. However, since it may not be immediately clear where the cited information is from[1], a reader has to invest additional effort to locate the cited information. We refer to (Wan, Paris, Muthukrishna, & Dale, 2009) for their survey results to justify this claims. In the series of surveys they conducted, most of their participants found it difficult to *find the exact text to justify the citation*. Quoting one of their participants' response directly: "*Citation usually does not include the position of the information* in the cited article... it might be necessary to read all of the article to find it in another reference and so on" (Wan et al., 2009).

[1] page numbers or references to specific artifacts, such as sections or equation numbers sometimes help to localize such references, but are not often included.

Citation Provenance refers to the source of a citation. The task of determining citation provenance is to locate the information in the cited paper that justifies the citation. It improves the reading experience of scientific and research documents by showing where exactly the cited information is from in the cited paper. We aim to identify which section or paragraph in the referenced paper is the cited information.

In this paper we describe a solution to this challenge: locating the information that justifies a citation. We will first look at some related works. In Chapter 3, we analyse the problem and describe observations made while building a corpus for training a supervised model of citation provenance. In Chapter 4, we discuss my approach on tackling the problem. We present our experimental results in Chapter 5, and then conclude.

Chapter 2

Related Work

Citation analysis is broad field of study, which has recently attracted computational methodology, using natural language and machine learning techniques for automation. We categorise such recent past works into several directions for development. A subfield of study that has a major impact is citation classification (similarly named as citation function). Such work aims to determine the basis for the authors' citation of the others' work, and thus better aid readers understand the key ideas presented in the paper. The reasons why authors would cite, are what was meant by the citation function. Teufel et al. (2006b) defined an annotation scheme (see Figure 2.1) for citation function that is able to describe the relationships between documents linked via citations.

Nakov, Schwartz, and Hearst (2004) discussed the potential of using text surrounding citations, *citances*, for automated analysis of bioscience literature. Dong and Schäfer (2011) presented an approach to citation classification in which, they extracted several features from *citances*. Some features worth mentioning are their *physical features*, that included the number of unique references cited within the *citances*, and one that measured the existence of cue words. Teufel, Siddharthan, and Tidhar (2006a) also described a similar feature that involved cue phrases, a strong indicator for citation function. Together, these previous works demonstrated the importance of utilising *citances* in citation analysis tasks.

In (Wan et al., 2009) and (Wan, Paris, & Dale, 2010), Wan and his team built a research tool that acts as a reading aid for readers when browsing through scientific papers. Wan et al. (2010) investigated the *literature browsing task* through surveys on researchers

Category	Description
Weak	Weakness of cited approach
CoCoGM	Contrast/Comparison in Goals or Methods (neutral)
CoCoR0	Contrast/Comparison in Results (neutral)
CoCo-	Unfavourable Contrast/Comparison (current work is better than cited work)
CoCoXY	Contrast between 2 cited methods
PBas	author uses cited work as starting point
PUse	author uses tools/algorithms/data
PModi	author adapts or modifies tools/algorithms/data
PMot	this citation is positive about approach or problem addressed (used to motivate work in current paper)
PSim	author's work and cited work are similar
PSup	author's work and cited work are compatible/provide support for each other
Neut	Neutral description of cited work, or not enough textual evidence for above categories or unlisted citation function

Figure 2.1: 12-class annotation scheme designed by Teufel et al. (2006b)

who read scientific papers frequently to keep up-to-date themselves. In the initial study conducted by Wan *et al.*, several key ideas were revealed. First, when researchers read scientific papers and see citations made by the author, their main concern – as time-constrained professionals – is whether the cited paper is worth their effort to follow up on. At the same time, the researchers need to know whether to believe the claim made in the citation. Second, readers faced the difficulty of finding the exact text that justify the citation. Third, the surveys revealed that readers thought that it would be useful if a reading tool could identify important sentences and key words in the cited paper. This study conducted by Wan et al. (2010) is based on the fundamental idea of improving the reading experience of researchers. The goal was to save a reader's time by assisting in the relevance judgement process on the cited documents. As it is often that readers

do need to read cited documents to gain insight on the current paper's context, this task is of relevance and importance. The authors then developed the *Context Sensitive In-Browser Summariser* (CSIBS) tool based on their studies. Figure 2.2 is an overlay of the CSIBS that displays citation-sensitive previews of the cited document. While it highlights matching keywords related to the citation, these sentences on the overlay do not necessarily justify the citation. To locate the provenance solely by word overlap would prove to be ineffective as paraphrasing and re-organising of sentence structure are common when authors cite previous works. There is a need to consider aligning *citances* to the cited document.

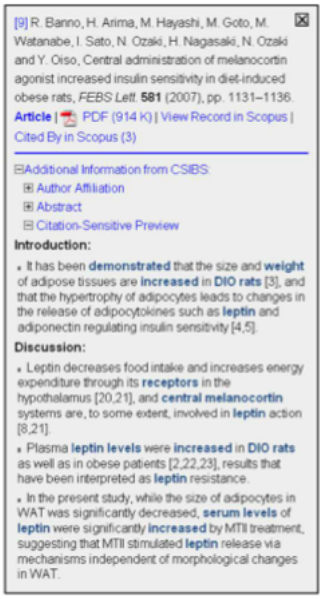

Figure 2.2: In-browser overlay preview of the CSIBS

Aligning sentences belonging to similar documents is an important research area for tasks related to summarisation and paraphrasing. Nelken and Shieber (2006) presented a novel algorithm for sentence alignment in for texts in a single language (i.e., monolingual corpora). They showed their approach, which is based on TF×IDF (a weighting scheme that reflects the importance of a word to a document in a collection of documents) similarity score, produced a high precision (83.1%) for the task of aligning sentence. Adding to what we mentioned early, authors paraphrase the content they were referring to usually for greater clarity and to introduce variety. While Shinyama, Sekine, and Sudo (2002) presented an approach to acquire paraphrase automatically, in our citation

provenance project, we aim for the converse goal. By comparing the words and phrases used in a citation with paraphrases extracted from a cited work, one may achieve improved sentence alignment between the two documents.

Chapter 3

Problem Analysis

For the scope of our project, we define all citations as belonging to one of two types: **General** and **Specific**. Specific citations refer specifically to particular parts or section in the cited paper, General citations do not. We use the following guideline to ensure that there is no ambiguity in our definition of the dichotomy between General and Specific:

General Citations

1. Authors may refer to a paper as a whole. If the author cites the cited paper for a key idea, e.g. Machine Learning, and Machine Learning makes up the entire or majority of the cited paper, it is a general citation.

2. Authors may refer to a paper as a form of mentioning. In such cases, the authors merely mention the cited paper to acknowledge the referenced authors for having done work on this particular research field.

Specific Citations

1. Authors may refer to a term definition in the cited paper.

2. Authors may refer to a key idea/implementation in the cited paper. This key idea or implementation does not constitute the entire or majority of the cited paper.

3. Authors may refer to an algorithm or a theorem in the cited paper. This algorithm/theorem and its supplementary evidence does not constitute the entire or majority of the cited paper.

4. Authors may refer to particular digits or numerical figures in the cited paper. This is usually done to make reference to quantitative evaluation results in the cited paper. Authors may also complement the cited paper for its performance.

5. Authors may quote a passage in the cited paper.

Term	Description
Citing Paper	The paper that makes reference to the cited paper
Cited Paper	The paper referred to by the citing paper
Cite Link	E.g. **E06-1034==>J93-2004**. A citation relation between a citing paper (**E06-1034**) and a cited paper (**J93-2004**). **E06-1034** is a sample ID for a scientific paper
Cite String	The in-line citation mark. E.g. Nivre and Scholz (2004), [1], (23)
Citing Sentence	A sentence in the citing paper that contains the cite string. E.g. *That algorithm, in turn, is similar to the dependency parsing algorithm of* ***Nivre and Scholz (2004)****, but it builds a constituent tree and a dependency tree simultaneously.*
Citing Context	*Citances* as defined in (Nakov et al., 2004). The block of text surrounding the citing sentence, about 2 sentences before and after the citing sentence, for providing contextual information
Cited Fragment	A fragment of text, from a few lines to paragraphs, in the cited paper

Figure 3.1: Terminological conventions used in this dissertation

For **Specific** citations, we need to extract a fragment in the cited paper that represents the source of the information mentioned in the citation itself, i.e. citation provenance.

3.1 Scope Of The Problem

We decompose the task into two tiers. In the first tier, we first determine whether a citation is General or Specific. If a citation is General, the reader can be directed, for

example, to the abstract of the cited paper. If a citation is Specific, the reader can be directed to the specific paragraph or line. From our definition, given that a citation is Specific, then there must exists a region in the cited paper that the citation refers to. To solve this second tier, we implement a ranking system that determines the location of this region.

Our project has a practical aspect that can be readily applied to scholarly papers. However, even with the components above, to field the project practically, the system must be able to detect in-line citations in a suitable textual representation of a scholarly paper corpus. While such engineering issues are important, our work focuses only on the research aspects of determining citation provenance, and hence we abstracted away the problem of locating in-line citations. Instead we reduced the problem to only determining the type of a citation and its location. To solve the practical problem of locating the in-line citations, we utilize the open-source logical document structure and reference string parsing system, ParsCit, developed in Councill, Giles, and Kan (2008). Conveniently, ParsCit identifies the citing sentence, together with its citing context.

3.2 Modelling The Problem As Search

In web search engines, an user enters a search query, and a search engine would use this query to search within its search domain – millions of web pages – and then display the best matching web pages as compared to the search query. In comparison for a citation, citing context is the query, the cited document is the search domain and the citation provenance is one (or more) of the 'web pages'. With the help of ParsCit (Councill et al., 2008), the citing context can be extracted. Our second subproblem of linking Specific citations to their origin can be modelled as search.

Cast this way, our first subproblem is simply a *binary classification problem*, where we attempt to determine whether a fragment is either General or Specific.

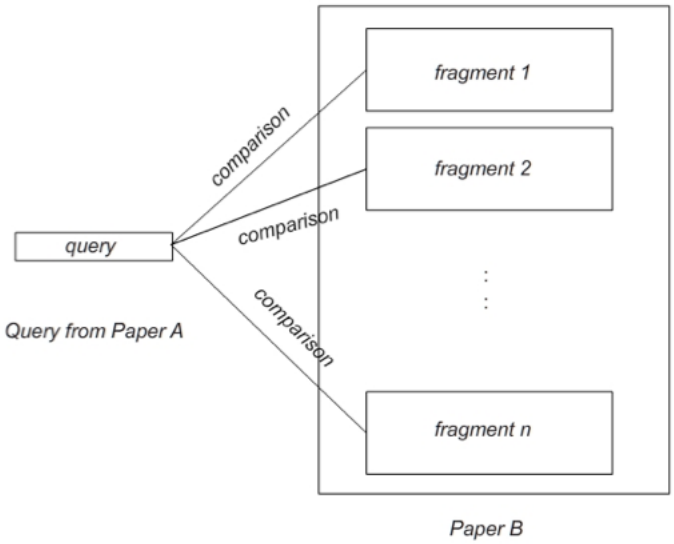

Figure 3.2: Modeling Our Problem

3.3 Target Corpus

We selected the ACL Anthology Reference Corpus[1] (ACL-ARC) as the target corpus to perform our research on. The ACL-ARC consists of publications covering topics in computational linguistics. While we wish to generalize our citation provenance methodology to work on publications from all fields of research, we chose to start with this corpus as it provides the *interlink data* that conveniently informs us of the cite links between the papers in the corpus. For instance, in the interlink data, a link like **X98-103==>X96-1049** says that the paper **X98-103**[2] cites **X96-1049**.

Given our formal problem statement, we can now specify the required data for the task. For each cite link, there can be multiple in-line citations i.e. multiple citing contexts, when a target paper is cited multiple times by the authors. Each citing context is compared with every fragment in the cited paper; i.e., if a cite link has n citing contexts and the cited paper can be divided into m fragments. This immediately gives rise to $(n \times m)$ data instances. Consider instance 3050 in Figure 3.3 that consists of a context-fragment pair. Note that a fragment is of arbitrary length. For each cite link, for each citing

[1] `http://acl-arc.comp.nus.edu.sg`

[2] All ACL-ARC papers are assigned an unique paper ID

context, there are m context-fragment pairs. One (or a few) of them is the correct pair that connects a citation to its origin.

Instance	Cite Link	Context	Fragment
⋮	⋮	⋮	⋮
3050.	X98-103==>X96-1049	...Penn Treebank 3 release (**Marcus et al., 1993**), the string in (1) is a variation...	...POS tagging phase is automatically parsed an...
3051.	X98-103==>X96-1049	...Penn Treebank 3 release (**Marcus et al., 1993**), the string in (1) is a variation...	...to annotate large corpora in the past hav...
⋮	⋮	⋮	⋮

Figure 3.3: Data instances

We need to obtain this information: Which context-fragments pairs connect its citation to its origin? For that we collect annotations for these citations.

Collecting Annotations – First Attempt

The first attempt at collecting annotations was to require an annotator to specify the line numbers of the cited information that the citing context was referring to. The annotator would be provided the citing and cited paper in plain text format, and he/she will need to annotate on a separate file, specifying the line number range, e.g. line range L12-55 of the cited paper. For this annotation task, we designed an annotation framework[3] where an annotator is presented with an user-friendly interface to select the lines in the cited paper that he/she deem Specific.

We planned for 2 means to collect annotations: One via inviting NUS students to computer labs to annotate these citations, the other via an online crowdsourcing solution, Amazon Mechanical Turk (MTurk[4]). The first allows close supervision, while the

[3] `http://citprov.heroku.com`

[4] `https://www.mturk.com`

second is an attempt to collect annotations on a larger scale. For collecting annotations from NUS students, we applied and were approved for review exemption from the NUS Institutional Review Board (IRB[5]) since our annotation task does not collect any confidential information from participants. After a trial round of annotation, we reviewed this annotation scheme together with feedback from the small group of participants.

First, this annotation task is non-trivial. Participants must be able to understand the contents of the papers, and thus, must largely be either subject matter experts (researchers) or have some experience in reading scientific papers. While it is possible to target a selected category of MTurk workers for this task, the complexity of this task requires participants with research experiences, which could be limited in numbers. Furthermore, most of the annotations collected from MTurk do not agree among the annotators and ourselves. Thus we abandoned collecting annotations via MTurk, and performed annotations manually.

Second, this annotation scheme is too tricky, and would also cause us much problem when it comes to evaluation. Consider an implemented system that outputs a prediction for citation provenance in the form of a line number range. It is difficult to judge the correctness of this prediction, say `L50-78`, when compared against the annotated `L12-55`. The prediction *overlaps* the annotation by 5 lines, but this variable amount of overlap is not definitive and difficult to decide at what extent of overlap only do we consider the prediction correct. Thus we switched to the alternative.

Collection Annotations – Second Attempt

The second attempt is more straightforward. Recall that we used ParsCit for extracting the citing context. ParsCit also divides a paper into logically adequate fragments according to sections, sub-sections, figures and tables etc. So instead of annotating the papers in plain text format by line number ranges, we annotated the structured output from ParsCit, each of the fragments of the cited papers with 3 classes: General (g), Specific-Yes (y) and Specific-No (n). To be precise, we annotated g (for all its fragments) if a cite link is deemed General, and y only for the fragment(s) that is deemed Specific.

[5] `http://www.nus.edu.sg/irb/`

For the other fragments that are not Specific, we annotated n. Table 3.1 summarises the statistics for annotation. Note that only percentage values for Specific instances are displayed.

Item	Statistics
No. of Cite Links	275 (7.6% Specific)
No. of Fragments	30943 (0.09% Specific-Yes, 12.9% Specific-No)

Table 3.1: Annotation Statistics

Specific citations are very rare and the training data is heavily skewed towards General citations. After prolonged periods of searching for valid Specific citations in our training corpus, we argue that despite more attempts to gather more positive instances, the ratio between General and Specific would remain the same. This challenging situation we have with the annotations also contributes to our approach to the problem, as we explain in the following chapter.

During the annotation process, we observed that Specific citations can be categorised into four sub-classes. We acknowledge that these observations are limited to the particular corpus we worked with and may not generalize. We observed that Specific citations may:

1. refer to digits/numerical figures in the cited paper, usually in the evaluation section
2. refer to term definitions by the author(s) of the cited paper
3. refer to algorithms/theorems in the cited paper
4. quote a line or segment in the cited paper

These observations also led to the implementation of some features that are defined next chapter in our approach.

Chapter 4

Approach

We describe how we adopt a supervised learning approach to tackle the problem that we had decomposed into two tiers.

4.1 *GvS* (First Tier)

GvS, short for General versus Specific, is the first tier in our approach. In *GvS*, we perform a binary *citation classification* task, which is already a challenging task. *GvS* makes use of information only from the citing contexts in a citing paper. We built a model based on features extracted from the citing contexts. With this model, *GvS* classifies citing contexts into one of the two classes.

Building The Model For *GvS*

GvS performs citation classification. Thus to build our model, we adopt past features used in Dong and Schäfer (2011) previously used for citation classification. In addition, based on the observations made during annotation, we introduced a few features that are targeted at Specific citations. From each of the 275 annotated cite links mentioned in Table 3.1 we extracted a set of features into a *feature vector* and map it to its *label* according to annotation (Figure 4.2). The features used are as described below. Note that the features with names beginning with the asterisk (*) are new features we introduced.

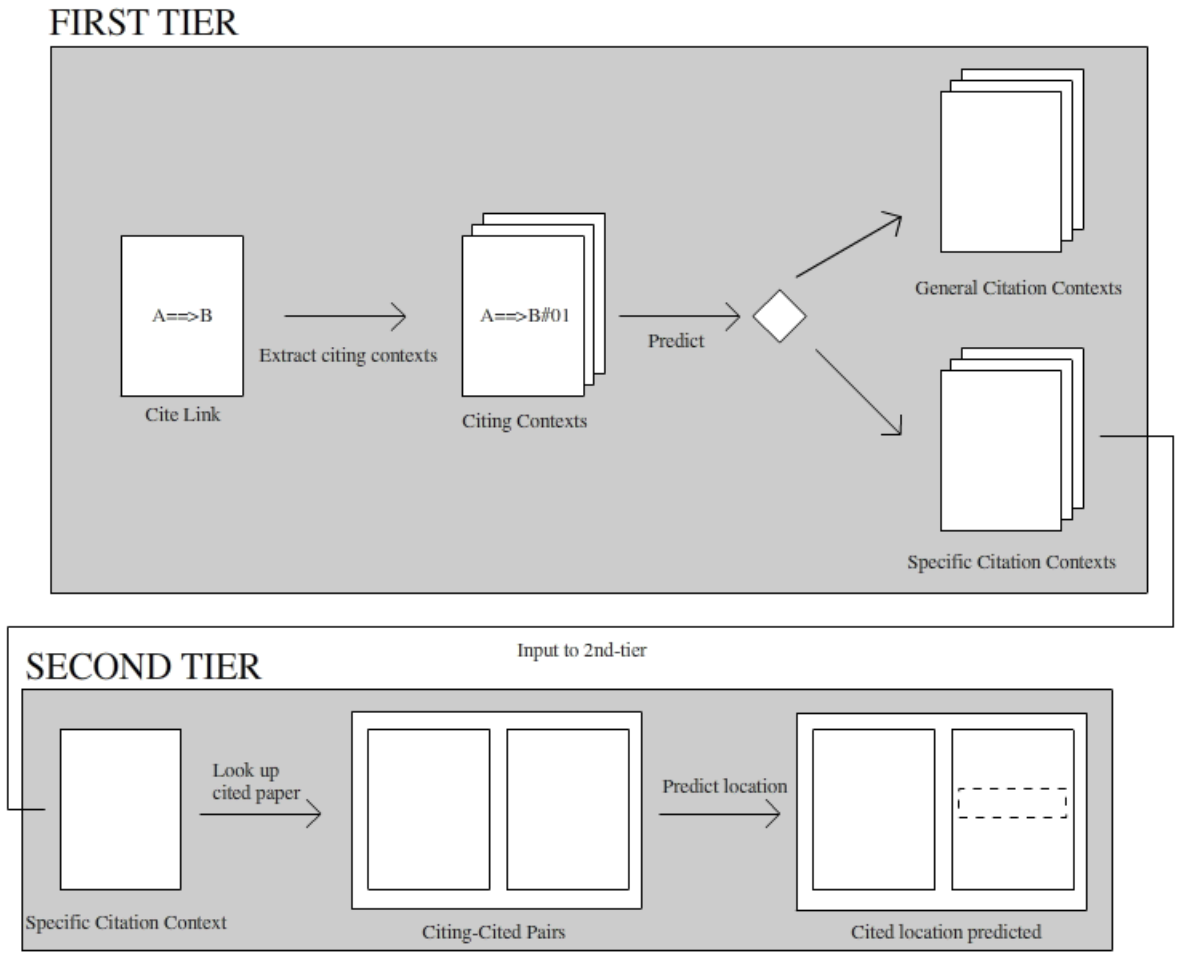

Figure 4.1: A Two-Tier Approach

GvS Features

1. Physical Features (Feature *A*)

   We adopted wholesale the physical feature set as presented in (Dong & Schäfer, 2011). They are:

   (a) *Location*: in which section the citing sentence is from.

   (b) *Popularity*: number of citation marks in the citing sentence.

   (c) *Density*: number of unique citation marks in the citing sentence and its neighbour sentences.

   (d) *AvgDens*: the average of Density among the citing and neighbour sentences.

$$
\begin{aligned}
v_1 &: [f_1, f_2, f_3, \ldots, f_n] \rightarrow L_1 \\
v_2 &: [f_1, f_2, f_3, \ldots, f_n] \rightarrow L_2 \\
&\vdots \\
v_i &: [f_1, f_2, f_3, \ldots, f_n] \rightarrow L_i \\
&\vdots \\
v_m &: [f_1, f_2, f_3, \ldots, f_n] \rightarrow L_m
\end{aligned}
$$

Figure 4.2: Mapping feature vectors to labels from annotation

The intuition for using these features is that: Based on our observations, citations found in the Evaluation section of the citing paper tend to cite results from the Evaluation section of the cited paper. Thus the *location* would suggest the type of citation. Also, General citations tend to have a higher number of citation marks within a citing sentence and its neighbour sentences. This is the rationale for the remaining 3 features.

2. *Number Density (Feature B)

   A numeric feature similar to the first feature set that measures the density of numerical figures in the citing context. This is based on our previously-described analysis in Chapter 3.3 that Specific citations tend to refer numerical figures in evaluation results in the cited paper. E.g. "...Nivre and Scholz (2004) obtained a precision of 79.1%...".

3. *Published Year Difference (Feature C)

   A numeric feature that represents difference in the publishing year between the citing and cited paper. A large difference means that a large publishing time gap exists between the citing and cited papers. Such long term citations are usually citations for General purposes.

4. *Citing Context's Average TF×IDF Weight (Feature D)

   A numeric feature that indicates the average number of *valuable* words, as determined by TF×IDF (Manning, Raghavan, & Schütze, 2008) in the citing context. Higher values suggest important, signature words pertinent to a specific claim. Thus high values of this feature we believe indicate a Specific use.

5. Cue Words (Feature E)

   Another numeric feature adapted from Dong and Schäfer (2011) that computes the count of specific cue words that appear in the citing sentence and its neighbouring sentences. We defined two classes of cue words: Cue-General and Cue-Specific (refer to Appendix A for list of cue words). These cue words were hand-picked, based on examples observed during the annotation process.

Recall that according to our annotation statistics, this task is heavily skewed towards General citations. Building a model based on this skewed set of data instances will produce a biased model that often predicts General. In fact, during some preliminary experiments where all data instances are fitted into the model, it outputs General for all its predictions. To address this problem, we train the model on artificially sampled *unskewed data*. From the set of labelled feature vectors, we first gathered the Specific instances. Then we **randomly** selected from the rest to have a $1 : 1$ of Specific vs. General instances. While this ratio appear unrealistic compared to the actual statistics, we argue that we are building a model using balanced data to measure its ability to differentiate between the two types of citation.

4.2 *LocateProv* (Second Tier)

LocateProv, short for Locate Provenance, is the second tier of my approach. The design of *LocateProv* is all its inputs are Specific citations predicts which of the fragments in the cited paper is the cited fragment. Resembling a search, in *LocateProv* the citing context becomes the *query* to match the cited fragments in the cited paper. To perform this ranking task, we import features that are prevalent that are basic mechanisms for ranking in Information Retrieval.

Building The Model For *LocateProv*

In *LocateProv*, we predict which cited fragment is the provenance of a citation. Instead of cite links, we used the annotated fragments in Table 3.1 to build the model. Unlike the first tier, the features used in the second tier are based on both the citing contexts

and the cited fragments. Similarly the feature vectors are mapped onto the annotated labels. Adopting the same notation as used in the first tier, in the list of features below, features with names beginning with an asterisk (*) are features we introduced.

LocateProv Features

1. *Surface Matching (Feature F)

   A numeric feature that measures the amount of word overlap between the citing sentence and a fragment in the cited paper.

2. *Number Near-Miss (Feature G)

   A numeric feature that measures the amount of numeric figures overlap between the citing sentence and a fragment in the cited paper. This feature will preprocess each fragment, rounding numbers or converting to percentage values when it tries to match similar numbers in the citing sentence. This feature was added because of the observations we made earlier in Chapter 3.3, that citations may refer to evaluation results in the cited paper.

3. *Bigram Matching (Feature H)

   A numeric feature that measures the percentage of bigrams overlap between the citing sentence and a fragment in the cited paper. This feature was added to preserve word order when comparing the citing sentence and the fragment. This feature was also targeted at Specific citations that refer to term definitions or quote directly.

4. Cosine Similarity (Feature I)

$$\text{cosine similarity} = \frac{v \cdot u}{|v||u|} \tag{4.1}$$

   A feature commonly used in information retrieval to measure similarity between the query and a candidate document. In our case, citing sentence and the fragment. v is a vector representation of the citing sentence and u is the fragment.

Most of these features are added based on some of the observations we made during the annotation tasks.

Recall that the data instances that were annotated are heavily skewed against Specific citations. In fact, the ratio of Specific-Yes instances compared to the rest is at least $1 : 1000$. It is impossible to train a model that is not biased with this entire set of instances. Hence we used the same method used in *GvS*: to use a $1 : 1$ of Specific-Yes vs Specific-No instances. Note that this also coincide with the design of *LocateProv* that inputs are only Specific citations. It was also not feasible to use the actual ratio between Specific-Yes and Specific-No because comparing a citing-cited pair of papers, the ratio of citing context to the number of fragments in the cited paper is easily $1 : 100$.

For both tiers, we trained the models using various classifiers and evaluated their performance. We discuss the evaluation process in the following chapter.

Chapter 5

Evaluation

We performed modular evaluations of *GvS* and *LocateProv*. For each tier we evaluated the performance of models trained with different classifiers using the same feature set as was described in the previous chapter. We examined the efficacy of Support Vector Machine (SVM), Naïve Bayes (NB) and Decision Tree (DT) learning models.

5.1 Evaluating *GvS*

Recall that we used a 1 : 1 of Specific versus General data instances for building the model. We evaluate *GvS* using the **Leave-One-Out** cross-validation strategy. In this strategy we leave one data instance out for testing while the rest are used for training and we repeat this for the number of instances. The main reason for using this strategy is because the number of data instances in the unskewed data set is already very small, and we wish to maximise them for training. For this strategy we compare the performance of the various classifiers, for each, computing the Precision, Recall and F_1 values.

	SVM			NB			DT		
Class/Values	P	R	F_1	P	R	F_1	P	R	F_1
general	0.76	0.79	0.77	0.64	0.82	0.72	0.67	0.64	0.65
specific	0.78	0.75	0.76	0.75	0.54	0.63	0.66	0.68	0.67

Table 5.1: Leave-One-Out Results for *GvS*

Examining the confusion matrix for the best performing SVM classifier below in Ta-

ble 5.2, we see that model yields almost identical amounts of false negatives and false positives, such that neither error class dominates. We conclude that the classifier thus far has a balanced performance.

	ACTUAL g	ACTUAL s
PREDICTED g	22	6
PREDICTED s	7	21

Table 5.2: Confusion Matrix for SVM with Leave-One-Out on *GvS*

We assessed the performance of *GvS* given varied amount of training data instances. We experimented on 3 variations of the amount of training data: 9%, 25%, 50% and 75%. The remaining percentage in each variation is used the testing. For this experiment we used the SVM classifier. Table 5.3 shows the results of this experiment.

Amount for Training	Accuracy
9%	0.294
25%	0.642
50%	0.714
75%	0.857

Table 5.3: Performance of *GvS* given varied amount of training data

From Table 5.3 we demonstrated the trend *GvS* showed improvement as the amount of training data increased. This shows the potential for better performance to be used in practice if provided a large enough training set.

We next assessed *GvS* by evaluating the important of individual features via a *feature ablation* study. In this feature ablation study, we use the SVM classifier. For each feature removed from entire set of features, we trained a classifier on the set of unskewed data instances. The rest of the features are used to train a model and then tested on the same set of data instances. To measure the performance each round, we used the conventional accuracy measure. Note that in Figure 5.1 the letters A to E represents the five features described in Chapter 4.1.

We observed that feature A (Physical Feature) has the most impact in the accuracy of the predictions, with the greatest drop in accuracy when A itself is removed. Using

Configuration	Accuracy
Full	0.911
Full − A	0.714
Full − B	0.875
Full − C	0.786
Full − D	0.911
Full − E	0.732

Configuration	Accuracy
Only A	0.696
Only B	0.589
Only C	0.625
Only D	0.535
Only E	0.696

Figure 5.1: Feature Ablation on *GvS*

A alone also results in one of the highest accuracy (see Figure 5.1). Feature D (Citing Context's Average TF×IDF Weight) appears to be the only redundant feature, but since it does not decrease the overall accuracy we shall include it nevertheless.

5.2 Evaluating *LocateProv*

We evaluate *LocateProv* using an identical set of experiments as was done to the first tier *GvS* classifier.

We evaluate *LocateProv* using the **Leave-One-Out** strategy together with various classifiers. Table 5.4 summarises the results.

	SVM			NB			DT		
Class/Values	P	R	F_1	P	R	F_1	P	R	F_1
Specific-No	0.92	0.82	0.87	0.84	0.96	0.90	0.89	0.89	0.89
Specific-Yes	0.84	0.93	0.88	0.96	0.82	0.88	0.89	0.89	0.89

Table 5.4: Leave-One-Out Results for *LocateProv*

The scores are very close to each other between the classifiers. Let us examine the confusion matrix from the Naïve Bayes classifier since it has the highest precision for classifying Specific-Yes instances, the class that we are most interested in obtaining high accuracy for.

LocateProv is aimed at identifying the Specific-Yes fragments in the cited paper. Our goal is to attain higher numbers in both the *g*-*g* and *s*-*s* cells in the confusion matrix.

	ACTUAL n	ACTUAL y
PREDICTED n	27	1
PREDICTED y	5	23

Table 5.5: Confusion Matrix for NB with Leave-One-Out on *LocateProv*

We achieved this in Table 5.5 and we can conclude that *LocateProv* has a promising performance in differentiating Specific-Yes (y) and Specific-No (n) fragments.

We next assessed the features added to *LocateProv* using the *feature ablation* strategy. Note that the letters F to I represents the features described in Chapter 4.2.

Configuration	Accuracy
Full	0.893
Full $-$ F	0.893
Full $-$ G	0.875
Full $-$ H	0.893
Full $-$ I	0.786

Configuration	Accuracy
Only F	0.714
Only G	0.625
Only H	0.607
Only I	0.875

Figure 5.2: Feature Ablation on *LocateProv*

From Figure 5.2 we can conclude that feature I (Cosine Similarity) remains to be the most important among the features for *LocateProv*. This is expected because as modelled in Chapter 3, *LocateProv* is a searching problem, thus an information retrieval technique is most applicable. Note that, however, these results is only this particular small test set, we cannot generalize that Cosine Similarity will work well in larger test sets.

For a more conclusive evaluation, we compare *LocateProv* to our baseline for this task. With *LocateProv* resembling a search problem, a feasible baseline is to compare the citing context with the fragments with Cosine Similarity, coupled with TF×IDF (Manning et al., 2008) weighting scheme. Essentially the baseline is just *LocateProv* running only on feature I (Cosine Similarity). For a fair comparison between *LocateProv* and the baseline, we artificially sampled a $1:1$ (Specific-No vs. Specific-Yes) training dataset as we did before to unskew the data instances. Specific-Yes instances were gathered, and the same number of Specific-No instances were **randomly** selected from the collection. For both *LocateProv* and baseline, they were trained (on 75%) and tested (on 25%) with

the SVM classifier. Note that the only difference between the data set is the random set of Specific-No instances. We compared their P/R/F values in Table 5.6.

	LocateProv			Baseline		
Class/Values	P	R	F_1	P	R	F_1
Specific-No	0.86	0.86	0.86	0.75	0.60	0.67
Specific-Yes	**0.86**	0.86	0.86	**0.80**	0.89	0.84

Table 5.6: *LocateProv* versus Baseline

Notice the precision values in bold in Table 5.6, that *LocateProv* attained a higher precision than the baseline. *LocateProv* performs slightly better at differentiating Specific-Yes fragments from Specific-No. Thus, justifying our approach to locating Specific-Yes fragments in the cited paper.

Chapter 6

Discussion

One of the main challenges we have with this task is the limited number of Specific citations in scientific papers. In the data set we built from annotation, we have 7.6% of Specific citations. As for the distribution of the types of fragments, we have only 0.09% Specific-Yes fragments. These fragments are the ones our task aims to identify.

We argue that even though the percentage of Specific citations is low and that the value of applications that perform such task seems low, citation provenance would prove to be an important reading tool that helps readers understand and navigate between papers that are linked via citations. We support our claim with evidence. We built a prototype application, CitWeb[1], that integrated our research work done on Citation Provenance. The application was built as an entry submitted to the **CodeForScience**[2] 2012 competition organised by Elsevier[3], targeted at building applications for ScienceDirect[4]. Our application was well received among the judging panel that consisted of professionals from fields related to information technology and libraries.

Regarding evaluation results, let us focus on *LocateProv*. The overall performance of our approach showed that our system only did slightly better than our baseline. This means, as a search problem it is, the features we added are not yet good enough for this task. It also suggests that for future work for this task, we must introduce better sentence alignment techniques and, for example, to explore paraphrasing.

[1] `http://www.youtube.com/watch?v=jgeEP9VdpqQ`

[2] `http://www.codeforscience.com/singapore`

[3] `http://www.elsevier.com/`

[4] `http://www.sciencedirect.com/`

Citation provenance, as our approach has shown, is essentially a combination of 2 mechanisms: Citation Classification and Search. *GvS* is our humble attempt to perform automatic classification of citations that is already a challenging task. We believe the integration of state-of-the-art classifiers could lead to significantly better performance in this task.

Chapter 7

Conclusion

In our thesis, we examine a new task in citation analysis, citation provenance. While Wan et al. (2010) presented the CSIBS tool that gave readers a preview of a cited paper, it does not provide information that justifies a citation. In our paper, we described the first attempt to provide a solution to the challenge of locating the origin of a citation

We presented a two-tier approach towards this problem, *GvS* and *LocateProv*. With the first acting as a filter to classify the citations into one of the two types: General and Specific. The second predicts which of the fragments in the cited paper are referenced by the citation. One of the difficulties we faced in this task is the highly unbalanced ratio between General versus Specific citations. Also, the annotation task is very challenging and would require experienced researchers who understands the content of the papers to be annotated. As a result all the training instances were manually annotated.

To train prediction models for this task, we artificially sample an unskewed set of instances, a balanced ratio of General versus Specific instances, and measured their ability to differentiate between the 2 types of citations. Feature analysis showed that most of the features are essential, with the Physical Features (Feature *A*) adopted from Dong and Schäfer (2011) proving to have the most discriminative power in *GvS*, and Cosine Similarity (a common mechanism used in Information Retrieval tasks) remained to be most important in *LocateProv*.

Finally, evaluations on *GvS* and *LocateProv* produced promising results in classifying General versus Specific citations and locating the cited fragment in the cited paper. *GvS* obtained an accuracy of 0.786 in our cross-validation, demonstrating its potential

to perform in practice. *LocateProv* showed an improvement compared to our baseline, justifying the features we introduced to target cited fragments.

References

Councill, I., Giles, C., & Kan, M. (2008). Parscit: An open-source crf reference string parsing package. *Proceedings of LREC*, Vol. 2008 (pp. 661–667), European Language Resources Association (ELRA), 2008.

Dong, C., & Schäfer, U. (2011). Ensemble-style self-training on citation classification. , 2011.

Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.

Nakov, P., Schwartz, A., & Hearst, M. (2004). Citances: Citation sentences for semantic analysis of bioscience text. *Proceedings of the SIGIR'04 workshop on Search and Discovery in Bioinformatics*, , 2004, 81–88.

Nelken, R., & Shieber, S. (2006). Towards robust context-sensitive sentence alignment for monolingual corpora. *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics* (pp. 161–168), 2006.

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

Shinyama, Y., Sekine, S., & Sudo, K. (2002). Automatic paraphrase acquisition from news articles. *Proceedings of the second international conference on Human Language Technology Research* (pp. 313–318), Morgan Kaufmann Publishers Inc., 2002.

Teufel, S., Siddharthan, A., & Tidhar, D. (2006a). Automatic classification of citation function. *Proceedings of EMNLP-06, Sydney, Australia*, 2006a.

Teufel, S., Siddharthan, A., & Tidhar, D. (2006b). An annotation scheme for citation function. *Proceedings of Sigdial-06, Sydney, Australia*, 2006b.

Wan, S., Paris, C., & Dale, R. (2010). Supporting browsing-specific information needs: Introducing the citation-sensitive in-browser summariser. *Web Semantics: Science, Services and Agents on the World Wide Web*, *8*(2-3), 2010.

Wan, S., Paris, C., Muthukrishna, M., & Dale, R. (2009). Designing a citation-sensitive research tool: an initial study of browsing-specific information needs. *Proceedings*

of the 2009 Workshop on Text and Citation Analysis for Scholarly Digital Libraries, NLPIR4DL '09 (pp. 45–53), Stroudsburg, PA, USA, 2009: Association for Computational Linguistics.

Appendix A

Cue Words

The following is the list of cue words used in one of our feature. During feature extraction, all words are stemmed before we make any comparison.

A.1 Cue-General

proposed, propose, presented, present, suggested, suggests, described, describe, discuss, discussed, gave, introduction, introduced, shown, showed, sketched, sketch, talked, adopted, adopt, based, originated, originate, built, researchers, comparative, comparison, following, previously, previous

A.2 Cue-Specific

obtains, obtained, score, scored, high, F-score, Precision, precision, Recall, recall, estimated, estimates, reported, reports, probability, probabilities, peaked, experimental, experimented, rate, error

Appendix B

Results Details (GvS)

B.1 Results: Leave-One-Out

	Precision	Recall	F_1-Score
g	0.76	0.79	0.77
s	0.78	0.75	0.76

	actual g	actual s
predicted g	22	6
predicted s	7	21

Table B.1: SVM $P/R/F_1$ Scores and Confusion Matrix

	Precision	Recall	F_1-Score
g	0.64	0.82	0.72
s	0.75	0.54	0.63

	actual g	actual s
predicted g	23	5
predicted s	13	15

Table B.2: Naive Bayes $P/R/F_1$ Scores and Confusion Matrix

	Precision	Recall	F_1-Score
g	0.67	0.64	0.65
s	0.66	0.68	0.67

	actual g	actual s
predicted g	18	10
predicted s	9	19

Table B.3: Decision Tree $P/R/F_1$ Scores and Confusion Matrix

Appendix C

Results Details (*LocateProv*)

C.1 Results: Leave-One-Out

	Precision	Recall	F_1-Score
n	0.92	0.82	0.87
y	0.84	0.93	0.88

	actual n	actual y
predicted n	23	5
predicted y	2	26

Table C.1: SVM $P/R/F_1$ Scores and Confusion Matrix

	Precision	Recall	F_1-Score
n	0.84	0.96	0.90
y	0.96	0.82	0.88

	actual n	actual y
predicted n	27	1
predicted y	5	23

Table C.2: Naive Bayes $P/R/F_1$ Scores and Confusion Matrix

	Precision	Recall	F_1-Score
n	0.89	0.89	0.89
y	0.89	0.89	0.89

	actual n	actual y
predicted n	25	3
predicted y	3	25

Table C.3: Decision Tree $P/R/F_1$ Scores and Confusion Matrix