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. In terms of $F_1$ score, our second tier (*LocateProv*), at 0.90, performed about 25% better than 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 hope this work also encourages meaningful discussions to designing a new citation style that better captures the provenance of the cited information. In the rest of this paper, 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. In an updated version of their paper, Teufel, Siddharthan, and Tidhar (2009) presented an annotation scheme for citation function. In their revised scheme, citations are generalised into four main categories: Weak, Contrast, Positive and Neutral. Some of these categories are further broken down into more specific sub-categories, producing a total of twelve classes for annotating citations. Teufel, Siddharthan, and Tidhar (2006) previously worked on the automatic classification of citation function, utilising features extracted from the *citing context*. Dong and Schäfer (2011) presented an approach to citation classification that uses a combination of various supervised learning algorithms. Similarly, authors worked on analysing the sentiment of citations to determine the polarity of these citations. Most recently, Athar (2011) used sentence structure based features extracted from the citing context to produce promising results.

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 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. While it helps reader determine whether to read on the cited papers by providing a contextual summary of the cited papers, it only provided a summarisation solution.

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 similarity score, produced a high precision (83.1%) for the task of aligning sentence. More recent work by Li, Sun, and Xue (2010) introduced a new sentence alignment algorithm called Fast-Champollion. Briefly, it splits the input text into alignment fragments and identifies the components of these fragments before aligning them using a Champollion-based algorithm.

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

In the scope of our project, all citations are classified into 2 types: **General** and **Specific**. We define citations as such to be inline with our goal. That is, to be able to tell if Specific, where the cited information is in the cited document. Otherwise, the citation would be deemed General. To rid of ambiguity in our definition of a General/Specific citation, we have the following guidelines:

#### General Citations

1. Authors may refer to a paper as a whole. If the author cites 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. The authors merely mentions the cited paper out of acknowledgement of its contributions.

#### 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/implementation does not make up the entire cited paper.

3. Authors may refer to an algorithm or a theorem in the cited paper. This algorithm/theorem does not make up the entire cited paper.

4. Authors may refer to digits or numerical figures in the cited paper. Usually for making reference to evaluation results in the cited paper. Authors may also complement the cited paper for its promising/excellent performance.

5. Authors may quote a line/segment in the cited paper.

| Term | Description |
|---|---|
| Citing Paper | The paper that makes the citation |
| Cited Paper | The paper that is being cited 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`) |
| Cite String | The citation mark. E.g. Nivre and Scholz (2004), [1], (23) |
| Citing Sentence | A sentence in the citing paper that contains the in-line citation. 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 | 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, from a few lines to paragraphs, in the cited paper |

Figure 3.1: Terminologies used in this paper

In general, for **Specific** citations, we specifically 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 reduced the problem to determining first whether a citation is General or Specific. If a citation is General, the reader can be directed, for example, to the Abstract section of the cited paper. If a citation is Specific, the reader can be directed to that specific paragraph or lines respectively. If given that a citation is Specific, then there must exists a region in the cited paper that the citation refers to. For this we needed to implement some ranking system that determines the location of this region.

We abstracted away the problem of locating the in-line citations in a paper, and reduced the problem to only determining the type of a citation and its location. To solve the problem of locating the in-line citations, we utilized the open-source ParsCit system developed by 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. That would be equivalent to having a search query for an entire corpus of research papers. This problem can also be modelled as a searching problem, but a reduced version as compared to web search engines.

Consider reading a paper, `A`. We know the citations made by `A`, and these cited papers are listed in its References section. From this our search domain for any query from `A` would be the contents of the list of cited papers. We reduce this search domain further when we are investigating a particular citation in `A`, say now paper `A` cites the paper `B`. Now, for this citation, the scope of search would be the sub-domain – contents of paper `B`. So instead of searching for the best matching document in the corpus, we are now searching within `B`. The search query is the citation from `A`, the *candidate documents* would be various regions (referred to as fragments) in `B` (Refer to Figure 3.2 for a simple illustration). With the help of ParsCit (Councill et al., 2008), the citing context can be extracted. The search query would be citing context which consists of the citing sentence.

Our problem is now a *binary classification problem*, where we attempt to determine whether a fragment is either General or Specific.

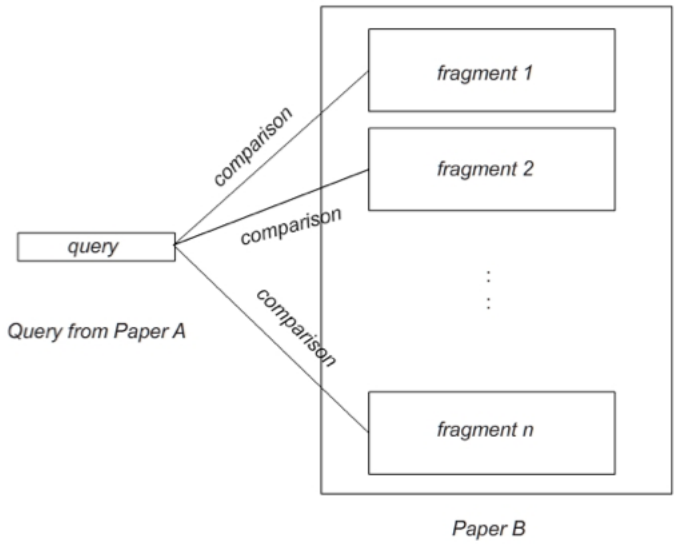


Figure 3.2: Modeling Our Problem

## 3.3 Building Our Corpus

At this initial stage, we picked the ACL Anthology Reference Corpus[1] (ACL-ARC). The ACL-ARC consists of publications of the Computational Linguistics field. Note that in general, we wish to perform this citation provenance task on all publications from all fields of research. This corpus is chosen as a start, because 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`.

Now that we have modelled our problem, we are able to specify the required data format for the task. For each cite link, there can be multiple in-line citations i.e. multiple citing contexts. Each citing context is compared with every fragment in the cited paper. In other words, if a cite link has $n$ citing contexts and the cited paper can be divided into $m$ fragments, immediately we have $(n \times m)$ data instances.

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

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

## 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 posted this task onto the Amazon Mechanical Turk (MTurk[4]) as an attempt to collect annotations on a larger scale and we collected some annotations from a few MTurk workers. After a trial round of annotation, we reviewed this annotation scheme together with feedbacks from the small group of participants.

First, this annotation task is a non-trivial one. Participants must be able to understand the contents of the papers, thus, must be 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.

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

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

### 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. 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 4 sub-classes. Note, however, these observations are for this particular corpus we worked with. 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 propose a two-tier approach to our problem. In the first tier, it plays the role of a *filter*, and attempts to filter out the General citations, leaving behind the Specific citations to be passed to the second tier. Figure 4.1 illustrates the flow of our approach.

## 4.1 *GvS* (First Tier)

*GvS*, short for General versus Specific, is the first tier in our approach to filter out the General citations. In *GvS*, we are performing a 2-class *citation classification* task, which already is a challenging task in the research area of citation analysis. We are not interested in determining whether the citation is one of the 12 class as defined by (Teufel et al., 2009), but only whether it is General or Specific. *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. Only those contexts that are classified as Specific will be passed to the second tier.

### Building The Model For *GvS*

To build a model to classify General versus Specific, we adopt some of the features that Dong and Schäfer (2011) used for citation classification. From each of the 275 annotated cite links mentioned in Table 3.1 we extracted a set of features into a *feature vector*

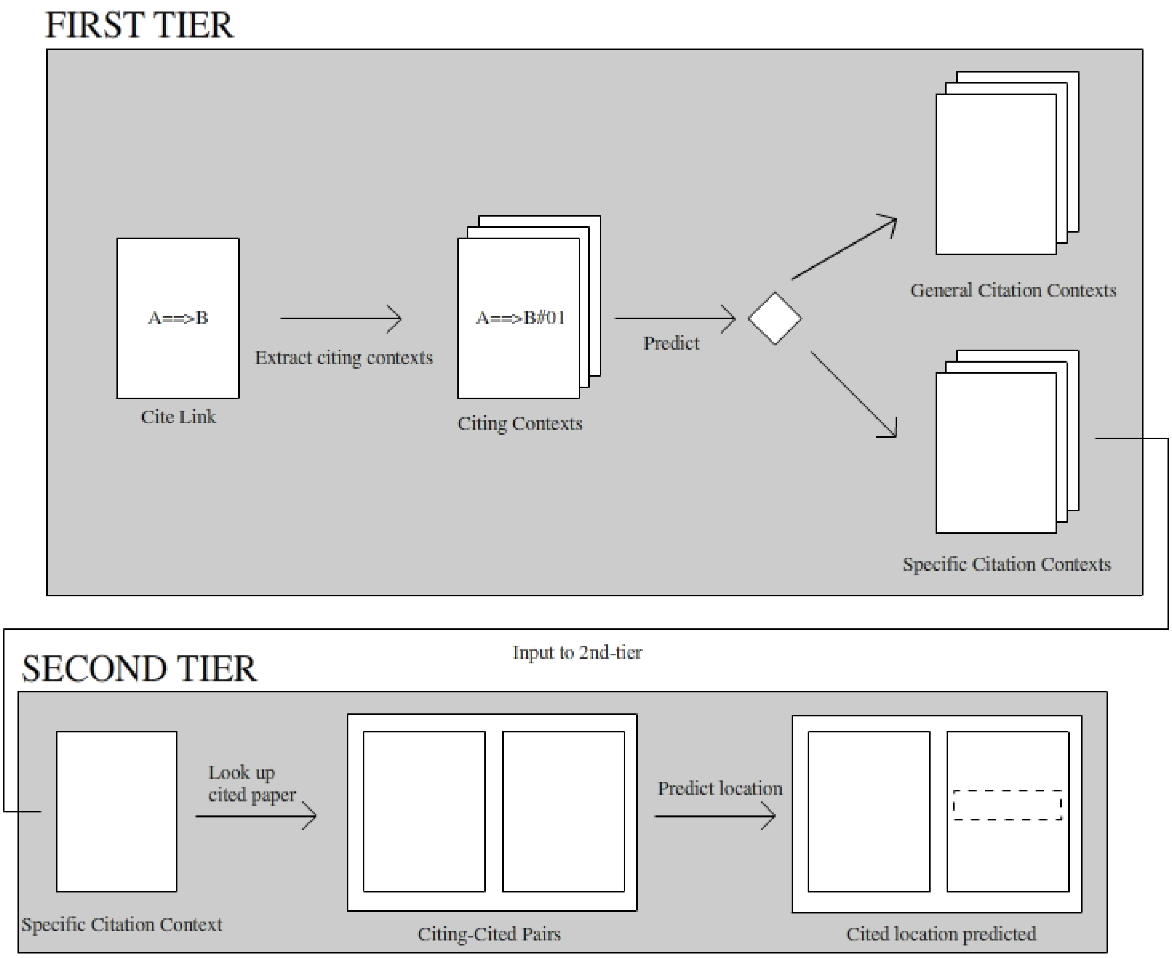


Figure 4.1: A Two-Tier Approach

and map it to its *label* according to annotation (Figure 4.2). The features used are as described below.

## *GvS* Features

1. Physical Features (Feature *A*)

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

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

   (b) *Popularity*: no. of citation marks in the citing sentence.

   (c) *Density*: no. of unique citation marks in the citing sentence and its neighbour

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

Figure 4.2: Mapping feature vectors to labels from annotation

sentences.

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

The intuition for using this feature is: A higher number of citation marks within a citing sentence suggests these citations are likely to be General since there was little room of discussion by the author(s).

2. Number Density (Feature $B$)

   A numerical feature similar to the first feature set that measures the density of numerical figures in the citing context. The intuition is that Specific citations tend refer numerical figures in evaluation results in the cited paper. E.g. "...Nivre and Scholz (2004) obtained a precision of 79.1%...". This feature was added based on the observations we made earlier in Chapter 3.3.

3. Publishing Year Difference (Feature $C$)

   A numerical feature that represents difference in the publishing year between the citing and cited paper. The intuition is that higher difference suggests cited paper is older and presented a fundamental idea, and thus cited for General purposes.

4. Citing Context's Average `tf-idf` Weight (Feature $D$)

   A numerical feature that indicates the average amount of *valuable* words, as determined by `tf-idf` (Manning, Raghavan, & Schütze, 2008), in the citing context. Higher values suggest important words and thus specific keywords.

5. Cue Words (Feature $E$)

   Another numerical feature adopted from Dong and Schäfer (2011), that computes the amount of cue words that appear in the citing sentence and its neighbour sentences. We defined 2 classes of cue words: Cue-General and Cue-Specific (refer to Appendix A for list of cue words). These cue words are hand-picked based on the examples we 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 only produce a bias 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 fix this problem, we proposed training the model on *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 am building a model using balanced data to measure its ability to differentiate between the 2 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. For that we also added some features that are common in Information Retrieval tasks.

### Building The Model For *LocateProv*

In *LocateProv* we are predicting 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. In this tier the features used are based on both the citing contexts and the cited fragments in order to *connect* the citation to its provenance. Similarly the feature vectors are mapped onto the annotated labels.

### *LocateProv* Features

1. Surface Matching (Feature $F$)

   A numerical 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 numerical feature that measures the amount of numerical figures overlap between the citing sentence and a fragment in the cited paper. This feature will preprocess each fragment, rounding numerical figures or converting to percentage values when it tries to match the numerical figures 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. Bigrams Matching (Feature $H$)

   A numerical feature that measures the amount 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 $I$)

   A common numerical feature used in Information Retrieval tasks to measure similarity between the query and a candidate document. In our case, citing sentence and the fragment.

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

Again, 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 bias with this entire set of instances. Hence we used the same method used in *GvS*: to use a 1 : 1 of Specific-Yes versus 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 performances on a few evaluation strategies. We discuss the evaluation process in the following chapter.

# Chapter 5

# Evaluation

We performed modular evaluation on *GvS* and *LocateProv*. For each tier we evaluated its performance on a few classifiers: Support Vector Machine (SVM), Naive Bayes (NB) and Decision Tree (DT). For each classifier we also performed evaluation using a few evaluation strategies.

## 5.1 Evaluating *GvS*

Recall that we used a 1 : 1 of Specific versus General data instances for building the model. To first verify *GvS*, we evaluated the features added using the *feature ablation* strategy. For each feature removed from this set of unskewed data instances, the rest of the features are used to train a model using the SVM classifier 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 and one of the highest accuracy when $A$ alone is used as a feature (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.

We first evaluated *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

| 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*

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 compared 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*

Let us examine the confusion matrix for the best performing SVM classifier that we ran for the `Leave-One-Out` strategy. *GvS* is aimed at filtering out the General citations. Our goal is to attain higher numbers in both the *g*-*g* and *s*-*s* cells in the confusion matrix. We achieved this in Table 5.2 and we can conclude that *GvS* has a promising performance in differentiating General and Specific citations.

| | 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 continued evaluating *GvS* using another cross-validation strategy, $K$-fold. While this appears to be a repeat usage of a cross-validation strategy for evaluation, we wanted to gain a better insight of *GvS*'s performance when given less training instances. For that we performed 7-fold cross-validation. From Table 5.3 we can conclude *GvS* could maintain promising performance in practice.

| $n^{th}$ Fold | Accuracy |
|---|---|
| 1 | 0.750 |
| 2 | 0.750 |
| 3 | 0.750 |
| 4 | 0.875 |
| 5 | 0.875 |
| 6 | 0.750 |
| 7 | 0.750 |
| Average | 0.786 |

Table 5.3: Cross-Validation on *GvS*

## 5.2 Evaluating *LocateProv*

Similar to evaluating *GvS* in Chapter 5.1, we first evaluated 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 solution is most suitable. Note that, however, these results is only this particular test set, which is also the training set. We cannot conclude that Cosine Similarity will work well in all cases.

We continue to 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 Naive Bayes classifier, which has the highest precision for classifying Specific-Yes instances.

| | 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*

*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. 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.

For a more conclusive evaluation, we compared *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 prepared 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 and tested on their own data set 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 | 0.96 | 0.82 | 0.88 | 0.89 | 0.57 | 0.70 |
| Specific-Yes | **0.84** | 0.96 | 0.90 | **0.61** | 0.89 | 0.72 |

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 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

Citation Provenance is a task that has little developments done on it. In this paper we defined the nature of the problem, and presented a possible approach to tackle it. One of the main challenges we had with this task is the limited number of Specific citations in scientific papers. Teufel et al. (2009) showed that the percentage of neutral citations was 62.7%. We can say that the percentage of General citations is at least as much, because our definition of a Specific citation is more restricted compared to the 12 classes defined by Teufel et al. (2009). This supports our observations during annotation collection that most citations are mere *mentions*.

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 with evidence the validity of our claim, that a prototype application (that performs Citation Provenance) submitted as part of the `CodeForScience`[1] 2012 competition organised by Elsevier was well received among the judging panel that consisted of professionals from fields related to information technology and libraries.

Sometimes, in-line citations to scientific papers in journals and books capture the chapter numbers and page numbers. The main reason is because the length of the cited document is very long compared to the citing document. An example of such citation is *(J. Doe, 2012, sec. 6.5, 174-85)*. In this citation it captures the section number, "*sec.*

[1] http://www.codeforscience.com/singapore

*6.5*", and page numbers, "*174-85*", to a book or journal. Note that the granularity of such style is not specific enough for our problem as a section can be arbitrary lengthy. In our case, we consider computational linguistic papers that are usually less than 20 pages, which is much shorter than books and journals. For this we sketch a new citation style that better captures citation provenance.

Our sketched style is straightforward: To numerically label each segment or fragment in the cited paper. This applies to text bodies, figures and tables. An example for a Specific citation: *(B. White, 2011, **B23**)*. Notice we added another a **B** to **23**, which could be a better way to distinguish between text bodies (B), figures (F) and tables (T). **23** simply means the 23rd segment of the type B. Suppose the cited paper is already labelled, when a reader sees a citation a paper, the reader sees there is the additional information at the end of the citation and understands it is a Specific citation. To read up on the cited paper would be a breeze.

# Chapter 7

# Conclusion

We touched on a new task for citation analysis, Citation Provenance. While Wan et al. (2010) presented the CSIBS tool that gave readers a preview of a cited paper, it only provided a summarisation solution. In our paper, we described the first attempt to provide a solution to the difficulty of locating the information that justifies a citation.

We presented a two-tier approach towards this problem, *Gvs* and *LocateProv*. With the first acting as a filter to separate the General citations from the Specific ones and the second one to predict which of the fragments in the cited paper are referenced by the citation. One of the challenges 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 gathered 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 strategy for 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. With that we conclude that we devised an approach to predict Citation Provenance.

# References


Athar, A. (2011). Sentiment analysis of citations using sentence structure-based features. *Proceedings of the ACL* (pp. 81–87), 2011.

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.

Li, P., Sun, M., & Xue, P. (2010). Fast-champollion: a fast and robust sentence alignment algorithm. *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 710–718), Association for Computational Linguistics, 2010.

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

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. (2006). Automatic classification of citation function. *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing* (pp. 103–110), Association for Computational Linguistics, 2006.

Teufel, S., Siddharthan, A., & Tidhar, D. (2009). An annotation scheme for citation function. *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue* (pp. 80–87), Association for Computational Linguistics, 2009.

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