Context Aware Citation Recommendation System

Hema Swetha Koppula
hsk49@cornell.edu

Abstract

This paper present a machine learning method for generating citation recommendations within research papers, articles, and other technical documents. This is potentially useful for authors who include some statement of fact within their work and go back only later to provide appropriate citation for such content. To facilitate this, we phrase this problem as an instance of multi-class learning in which we map so-called 'local contexts' in a paper to possible documents that would best serve as a cited source. Within this paper, we look at the effectiveness of a support-vector machine classifier alongside a feature set consisting of a mix of bag-of-words document descriptions and similarity measures between documents, and the difficulty of scaling such techniques given the massive amounts of data available and necessary for proper classification. We closely analyze specific results over a subset of this data taken from $Citeseer^X$.

Keywords: Context, Extraction, Recommendation Systems

1 Introduction

While writing research papers, we often want to find good citations for various snippets of text in the paper. To do this, one needs to search relevant literature and find a small number of good citations. This is a very time-consuming task when the amount of literature present on the topic is very large or when one is new to the area of research. It would be very useful to have a citation recommendation system which can suggest a small number of good citations given a snippet of text which needs to be cited. Given the large number of research papers with examples of good citations, we seek to build a supervised learning algorithm which learns a model to suggest good citations given a context for citation.

2 Problem Definition

For a document to be cited, the text surrounding a citation is considered the $localcontext$ of a citation. The text from the other parts of the document such as title, abstract are considered the $globalcontext$ of the paper. For classification, we used a multi-class SVM approach where each cited paper in our training set is considered a class. The features for classification are words from the local context, the global context and cosine similarity between the test document and possible documents to be cited (papers that are a class label).

For each test document, we return a ranked list of k possible citations (class labels).

3 Related Work

Previous approaches mostly deal with recommending a bibliography list for a manuscript or recommending papers to reviewers. They rely on a user profile or a partial list of citations. The most recent approach by He et al. [4] uses a context-aware approach for recommending citations. It uses the following methodology: A candidate set of papers are selected based on heuristics such as has an author in common, most similar abstract, most similar title, etc. These papers are then scored by measuring the similarity between the manuscript and the candidate. They propose a non-parametric probabilistic model to measure the context-based and overall relevance between the two documents. Tang and Zhang [2] uses a topic based recommendation approach for recommending citations. It does so by modelling the topic distributions of paper and their cited documents using a Restricted Boltzmann Machine (RBM) model. For each document to be cited it will model the topic representation of the paper and use a probability measure to compute how likely a paper be cited and return the top K papers with the largest probability. In recommending local context, they compute the KL-divergence of the local context with the recommended paper.

4 System Architecture and Implementation

Generating training data for our classifier proved fairly straight forward, though also computationally intensive given the amount of data available. Raw data was taken from $Citeseer^X$ in the form of plaintext representations of papers. While this had the effect of rendering unreadable most notation, charts, and diagrams, the actually body text of the papers remained more or less intact. Using the same processing tools employed by $Citeseer^X$ itself, we proceeded to extract citation information from our chosen data set. This consisted of local contexts surrounding the citation, as well as the titles, authors, and publication dates of the cited papers. This could be matched against $Citeseer^X$ metadata to determine unique identifiers for most cited papers in the dataset. Additionally, these tools also served useful in parsing apart the body text of input papers into its constituent sections, allowing us to use paper abstracts, section headers, authors, etc. as the basis of additional training features should it prove necessary.

In pre-processing our documents, we tokenized words by space characters, only extracting words consisting of letters (after converting all uppercase letters to lowercase ones), dashes, underscores, and apostrophes. We also discarded the top 1000 words according to word count from all documents, words with a length ¡ 3 and words with word count ¡ 3. Then for each document, we constructed a word histogram of the global and local context.

Document frequency and word frequency (total number of times a given word appeared in the data set) were calculated with a script that analyzed the entire data set calculated frequencies for each word in the dataset.

The word histograms were also used to compute cosine similarity scores between pairs of documents. For the purpose of readability and debugging these results were stored in plaintext as a sparse matrix. The size of the result was kept fairly reasonable under our restriction of only taking the top 100 matches for each document, and running time was greatly improved by first computing an inverted index over the data set, freeing us from the need to calculating all possible pairs.

5 Methodology

Every citation in a research paper can be considered as an input example for the learning algorithm. For a given citation, the words surrounding a citation can be considered as the local context and words from other parts of the paper, for example: title, abstract, introduction, etc., can be considered as the global context. We consider each cited paper as a class label. The feature set include words from the local context and global context. Multi-class classification can be used to learn the class labels for the given input features generated from the papers. We used $SVM^{multiclass}$ [4] for the multi-class classification. It uses the formulation described in [11] for multi-class classification. To solve this optimization problem, $SVM^{multiclass}$ uses an algorithm based on Structural SVMs [3].

Structural SVM tries to learn a discriminant function $F : X \times Y \rightarrow \Re$ over input/output pairs from which we can derive a prediction by maximizing F over the response variable for a specific given input x. Here F is assumed to be linear in some combined feature representation of inputs and output $\Psi(x, y)$. For a training set $(x_1, y_1)...(x_n, y_n)$ with labels y_i in $[1..k]$, it finds the solution of the following optimization problem during training.

$$\begin{gathered}\min_{w,\xi>0} \tfrac{1}{2} w^T w + C\xi, \\ s.t. \forall(\bar{y_1}, ..., \bar{y_n}) \in Y^n : \tfrac{1}{n} w^T \sum_{i=1}^{n} [\psi(x_i, y_i) - \psi(x_i, \bar{y_i})] \geq \frac{1}{n} \sum_{i=1}^{n} \Delta(y_i, \bar{y_i}) - \xi\end{gathered} \tag{1}$$

C is the regularization parameter and $\Delta(y_i, \bar{y_i})$ is the loss function which returns 0 if y_i equals $\bar{y_i}$, and 1 otherwise.

For multi-class classification a joint feature map is computed as $\Psi(x, y) \equiv \Phi(x) \otimes \Lambda^c(y)$, where $\Phi(x)$ denoted the original feature vector, Λ^c denotes the binary encoding of label y and $\otimes$ is the tensor product. The vector $\psi(x, y)$ can be seen as the feature vector x stacked into position y:

$$\psi_{multi}(x, y) = \begin{pmatrix} 0 \\ . \\ . \\ 0 \\ x \\ 0 \\ . \\ . \\ 0 \end{pmatrix}$$

Masks: The above definition of $\psi(x, y)$ leads to a very high number of terms in $\Psi(x, y)$ and is not scalable when we have large number of features and large number of classes. To improve scalability, we define a *Mask* for each target document as the bag of words which represent the document. We pick the top K words based on their Information Gain for class c_i as the *Mask* for c_i . The Information Gain of term t for class c_i is computed as follows:

$$G_{c_i}(t) = \sum_{X \in \{t, \bar{t}\}} \sum_{Y \in \{c_i, \bar{c_i}\}} Pr(X,Y) log \frac{Pr(X,Y)}{Pr(X)Pr(Y)}$$

These *masks* are used in order to specify that the classifier for class c_i only depends on the features specified by $mask_i$. This reduces the feature space $\Psi(x, y)$ as only a small number of features for each class will be considered. While calculating $\psi(x, y)$, the feature vector x is first masked with $Mask_y$ and then shifted to position of y. We made changes to the $SVM^{multiclass}$ to incorporate these *masks*.

Similarity Features: Other than just considering the words of the documents, we also use similarity features which measure how similar the citing document is to the cited document. These features include: Jaccard Similarity coefficient of source and target documents with the following combination of words: global context of source document and full target document, local context of source document and full target document, local context of source document and title of target document, local context of source document and abstract of target document, etc. Let $sim_i(x, y)$ represent the i^{th} similarity score, and l be the total number of similarity features. Including the masks and similarity features, $\psi(x, y)$ will look like this:

$$\psi_{citerec}(x, y) = \begin{pmatrix} 0 \\ . \\ . \\ 0 \\ Mask_y(x) \\ 0 \\ . \\ . \\ 0 \\ sim_1(x, y) \\ . \\ . \\ sim_i(x, y) \\ . \\ . \\ sim_l(x, y) \end{pmatrix}$$

In order to learn the finer dependency of the relevance of the citation on the similarity features, we use binning of the similarity score features. This is a tunable parameter in our experiments.

6 Experimental Evaluation

6.1 Data

Research papers from $CiteSeer^X$ [5] database were used to generate the training and test data sets. The research papers in text format were downloaded from $Citeseer^X$, and the metadata associated with each paper was obtained using the OAI harvesters. The text data of the papers was parsed to extract the context of each citation and the cited paper. The context includes various fields such as the local context, global context, introduction, abstract, conclusion, authors and title. The papers follow different citation formats, therefore each citation is resolved to an unique identifier by matching records from metadata.

Data	Description	#Labels	Train Data Size	Test Data Size
Set1	Rarely Cited	1000	3292	823
Set2	Medium Cited	1000	13142	3286
Set3	Most Cited	1000	50664	12668

Table 1: Data characteristics

The distribution of the number of citations each paper receives follows he zipf's law. In order to evaluate the performance of our approach on documents belonging to various regions of the distribution, we sub-sampled all the citation records into three data sets. The first set contains the rare citations, second has cited documents from the mid-frequency range and the third set comprises the most frequent citations. For the rare documents we take citations which are cited at least twice. The data characteristics are described in Table1. We randomly sample 20% of the data for the test set, and use the rest for training.

We considered six combinations of features sets as described below:

- Local context : Words from the local context as features with tf-idf scores as feature values
- Local context with Similarity Features : Local context features along wiht the similarity features
- Global context : Words from the global context as features with tf-idf scores as feature values
- Global context with Similarity Features : Local context features along with the similarity features
- Local and Global context : Words from local context and top 200 words from global context based on tf-idf score as features with tf-idf scores as feature values
- Local and Global context with Similarity Features: Local and Global context along with similarity features

The word feature set includes all unique words which appeared in the set of source and targets documents, which is about 272K words for our data set. The feature values are the *tf-idf* weights for that feature (word) in the corresponding context. The similarity features include the cosine similarity of the whole source and target document, and the cosine similarity of the local context and target document.

6.2 Results

We use 4-fold cross-validation on the training data to find the best value of C for each dataset and feature set combination. The final model is trained on the full training set and we report the performance on the test set. We evaluate the performance of the classifier and how it changes with the various input feature sets described in Section 6.1. We also evaluate how performance changes with the number of words selected as Masks. Finally, we evaluate our system based on the classification accuracy as well as retrieval accuracy, i.e., the recall of the system on selecting the top k results based on the classification score.

6.2.1 Effect of different feature sets

We trained our classifier on each of the three data sets with different input feature sets described in Section 6.1. Table 2 shows the train and test errors for the various features on dataset Set1. The Global Context with similarity features gives the lowest test error.

Feature Set	Train Error	Test Error
Local Context	11.82	46.66
Local & Similarity	13.46	29.64
Global Context	3.95	27.33
Global & Similarity	3.71	26.61
Local & Global Context	4.13	39.73
Local & Global & Similarity	3.88	37.91

Table 2: Effect of various feature sets on performance on Set1

Table 3 shows the train and test errors for al the feature sets on dataset Set2. The Global Context features gives the lowest Test error even thought the Train error is least for Global Context with similarity features. It can be seen from both Table 2 and Table 3 that the Similarity features do not have much impact when used with Global Context. This is because when all the words in the document are given as features, the classifier is able to learn the similarity with target documents even without the help of similarity features.

Feature Set	Train Error	Test Error
Local Context	27.26	42.88
Local & Similarity	22.46	35.74
Global Context	18.49	34.34
Global & Similarity	17.23	35.19
Local & Global Context	21.82	44.71
Local & Global & Similarity	28.08	43.76

Table 3: Effect of various feature sets on performance on Set2

Table 4 shows the train and test errors for the feature sets on dataset Set3. Here the best set of features are Local Context with similarity features. The Global Context features do not perform well. In the case of most frequent target classes, considering global contexts of all the documents citing a target paper will tend to introduce noise. Therefore, it makes more sense to only consider the Local Context features for most frequent labels. Including the similarity features along with the Local Context features reduces the test error by 5.3%.

Feature Set	Train Error	Test Error
Local Context	49.43	56.92
Local & Similarity	47.30	53.88
Global Context	53.40	62.27
Global & Similarity	49.63	58.75
Local & Global Context	64.07	72.43
Local & Global & Similarity	53.67	63.89

Table 4: Effect of various feature sets on performance on Set3

6.2.2 Effect of Mask Size

For measuring the effect of the mask size, we used Masks of 50 and 100 words and compare the Test Error and Training Time on all the three data sets. Table 5 shows the comparison on the three datasets when Local Context features are used and Table 6 shows the same for Global Context features. It can be seen that increasing the mask size from 50 to 100 reduces the test error. There is also an increase in the training time with increase in mask size, showing a trade off between getting better accuracy versus saving training time.

Data	Mask Size	Test Error	Training time
Set1	50	46.66	78.36
Set1	100	45.20	64.23
Set2	50	42.88	236.70
Set2	100	38.78	285.53
Set3	50	56.92	779.88
Set3	100	53.73	1003.05

Table 5: Effect of mask size with Local Context features

Data	Mask Size	Test Error	Training Time (in sec)
Set1	50	27.33	920.41
Set1	100	24.05	3093.01
Set2	50	34.34	7391.05
Set2	100	32.27	28603.36
Set3	50	62.27	24708.82
Set3	100	61.46	94881.87

Table 6: Effect of mask size with only Global features

6.2.3 Ranking Evaluation:

The test error and train error reported in the previous sections consider only the highest scored label and marks it as an error if it not the correct citation. However, we would like evaluate a list of high scoring documents returned by our system. To evaluate the ranking, we calculate the recall metric. It is the % of number of times the correct citation is present in the top-k scored documents returned by the classifier. Table 7 and 8 show how the recall changes with k in all three datasets and for all six feature sets.

	Local Features			Global Features			Local & Global Features		
k	Set1	Set2	Set3	Set1	Set2	Set3	Set1	Set2	Set3
1	53.34	57.12	43.08	72.66	65.66	37.73	60.27	55.29	27.56
5	70.96	76.40	67.14	85.18	84.09	64.37	77.04	73.69	47.67
10	76.55	81.48	73.77	88.46	87.89	74.04	82.14	79.04	56.21
15	78.49	83.30	77.06	89.67	90.60	78.69	83.84	81.93	61.04
20	79.95	85.16	79.06	91.37	92.00	82.00	85.42	83.76	64.38

Table 7: Ranking performance (Recall) with word features

	Local Features			Global Features			Local & Global Features		
k	Set1	Set2	Set3	Set1	Set2	Set3	Set1	Set2	Set3
1	70.35	64.26	46.12	73.39	64.81	41.25	62.09	56.23	36.11
5	87.12	84.00	71.86	85.66	82.57	68.83	79.95	74.66	62.74
10	93.07	89.39	79.24	88.82	88.02	78.48	84.33	81.20	72.83
15	94.90	91.60	82.75	90.77	90.60	83.09	87.12	84.25	77.66
20	96.11	93.19	85.42	91.85	91.88	85.96	89.19	86.49	80.88

Table 8: Ranking performance (Recall) with word and similiarity features

Figures 1,2 and 3 show how recall changes with k, where k is the number of top ranked results, for the data sets Set1,Set2 and Set3 respectively. It can be seen that when we consider the top 10 documents returned by our system, the model trained on Local Context with Similarity features gives the best recall on all the three data sets.

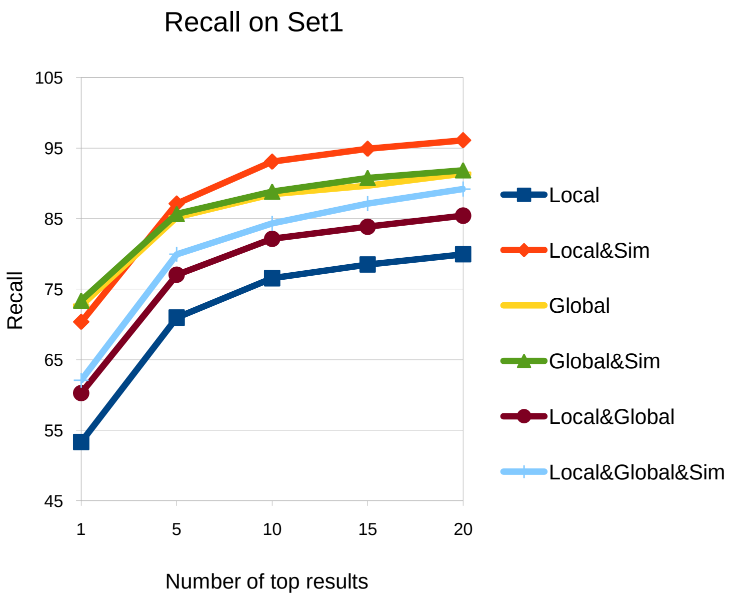

Figure 1: Ranking performance on rare target documents

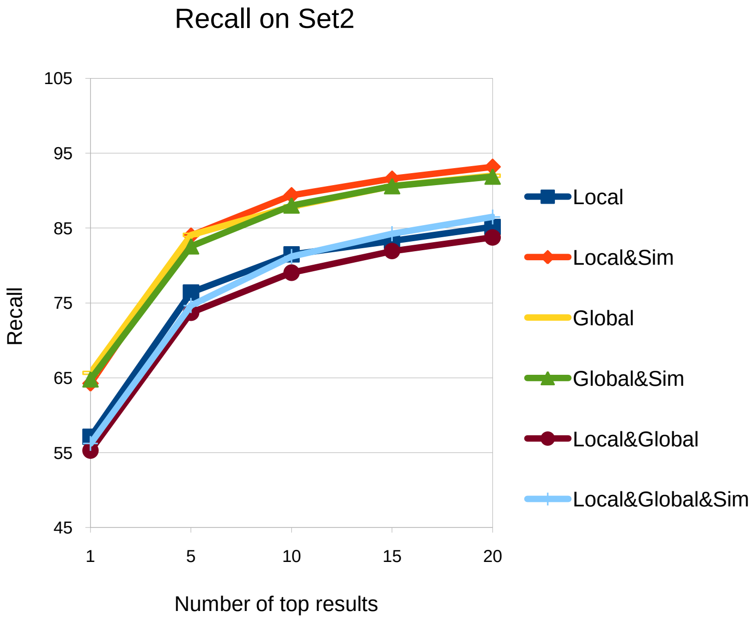

Figure 2: Ranking performance on medium frequent target documents

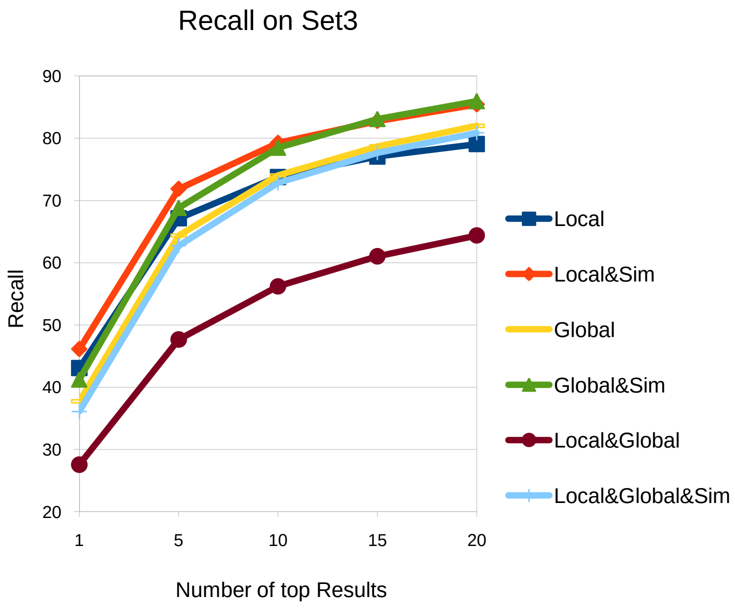

Figure 3: Ranking performance on most popular target documents

6.3 Comparison with Baseline:

Baseline: We use the ranked list of target documents based on cosine similarity between the local context and the target document. The words are weighted by the it-idf score of the word in the local context and the document respectively.

Table 9 shows the comparison of recall values between the baseline method and our model trained on Local Context with Similarity features. For all values of k, our approach performs better than the baseline. Its can be seen that for low values of k the performance gap is higher than at higher values of k. This shows that the our method outperforms the baseline in general and also is better at ranking the correct citations higher.

Data Set	k = 1		k = 5		k = 10	
	Baseline	Local & Sim	Baseline	Local & Sim	Baseline	Local & Sim
Set1	57.10	70.35	84.20	87.12	89.67	93.07
Set2	50.15	64.26	78.86	84.00	86.77	89.39
Set3	31.55	46.12	63.68	71.86	75.48	79.24

Table 9: Recall comparison with Baseline

7 Future Work

Ideally, the eventual result of this work would come in the form of a freely available online service to anyone wishing to use it. This is still some way in the future, however, and a great deal of preliminary work still needs to be done. The first steps going forward will likely be addressing the issues involved in scaling the method to deal with larger data sets. While we are very pleased with our results over our set of test data, it would be prudent to make sure we can offer similar results for a full set of documents that one might be using under a practical application of this system before making any other radical changes to the learning process. Assuming we could deal with these issues, or be reasonably assured that these issues could be efficiently dealt with in the future as processing power becomes more efficient, it would be logical to begin experimenting with variations on our current choices of features and training methods. Our current approach may seem somewhat simplistic in its reliance primarily on bag-of-words type descriptors of the data, and effectively extending this work to a larger set of documents may require more varied and complex ways of differentiating documents.

There might also be some utility found in looking at different ways of generating and evaluating our results. Our current assumption is that best set of results will be ones similar to those that would be found by a human, but this may not necessarily be the case. Humans are influenced heavily by social trends, and there is an argument to be made that the more recent and more frequently cited papers will continue to be cited more often (rich-get-richer phenomenon) because people are used to seeing them rather than for any inherent merit over the alternatives. Our results should not necessarily be held sway to these same trends just because those are what appear in the training data. Consequently, a beneficial line of future work might be to look at this as a diverse ranking problem, offering users a set of good matches that are also somewhat different from each other, rather than perhaps a number of very similar documents all from the same author or institution.

8 Conclusion

Any such system based on this or other work will have to take great care in accounting for the scale of the data involved and the computation time needed to retrain as new data is added. Even working over a small subset of the data available, it took several hours for our group to train our classifier. This time will only increase when run over all available data, and it would be impractical to retrain the entire system for every new document encountered. In addition to the obvious practical concerns of maintaining an up-to-date service, if we take this fact to be unavoidable under the current restrictions on processor speeds, it is necessary to consider the fairness of not immediately including new contributions to research in the system, and how it might affect the distribution of citations should such a system as this enter into wide spread use.

In this work we have presented a machine learning method based on Structural SVMs for doing citation recommendation. We have evaluated our approach on three different datasets representing three categories of citations, namely popular citations, regular citations and rare citations based on how frequently they are cited. In order to deal with the large number of targets, we introduced Masks for each target class and showed the trade off between test accuracy and training time while choosing the mask sizes. We also evaluated various feature sets and showed that using Local Context with Similarity features gives the best test accuracy when top 10 documents scored by the systems are considered.

9 Acknowledgments

- This work is done in collaboration with Dane Wallinga, Jun Erh and Sarah Nguyen

- Thorsten Joachims, Cornell University: Advisor
- Katharina Morik, Technische Universität Dortmund: Consulting and suggestions for future work
- Martin Berggren, Cornell University: Systems support

10 References

[1] Crammer, K. and Singer, Y. 2002. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res. 2* (Mar. 2002), 265-292.

[2] J. Tang and J. Zhang. A Discriminative Approach to Topic-Based Citation Recommendation. *PAKDD'09*.

[3] Tsochantaridis, I., Hofmann, T., Joachims, T., and Altun, Y. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the Twenty-First international Conference on Machine Learning*(Banff, Alberta, Canada, July 04 - 08, 2004). ICML '04, vol. 69. ACM, New York, NY, 104.

[4] He, Q., Pei, J., Kifer, D., Mitra, P., and Giles, L. 2010. Context-aware citation recommendation. In *Proceedings of the 19th international Conference on World Wide Web* (Raleigh, North Carolina, USA, April 26 - 30, 2010). WWW '10. ACM, New York, NY, 421-430.

[4] http://svmlight.joachims.org/svm_multiclass.html

[5] http://citeseerx.ist.psu.edu/