Extracting Useful Information from the Full Text of Fiction

Sharon Givon[#] & Maria Milosavljevic[*]

[#]School of Informatics
University of Edinburgh
2 Buccleuch Place, Edinburgh EH8 9LW, UK
S.Givon@sms.ed.ac.uk

[*]Centre for Language Technology
Macquarie University
Sydney, NSW 2109, Australia
mariam@ics.mq.edu.au

Abstract

In this paper, we describe some experiments in large-scale Information Extraction (IE) focusing on book texts. We investigate the scalability of IE techniques to full-sized books, and the utility of IE techniques in extracting useful information from fiction. In particular, we evaluate a variety of Named Entity Recognition (NER) techniques in identifying the central characters in works of fiction. First, we describe the creation of a gold standard for evaluation, which contains ordered lists of characters for a corpus of classic book texts in Project Gutenberg. Second, we describe several approaches to the task of character identification, where our best model achieves an average coverage score of 78.4% across all central characters. Finally, we propose a number of approaches for future work.

Introduction

Recent interest in the full text of books from major players in the text industry, such as Amazon.com and Google, has fuelled speculation about what might be done with such a corpus (e.g. Crane, 2006). There are two easily noticeable differences between a corpus of books and the traditional corpora commonly used in the field of Information Extraction (IE) to date. First, book texts are several magnitudes of size larger than the more commonly studied newswire articles or blog posts. Second, the nature of books is extremely diverse (ranging from children's fiction to mathematical proofs, for example). Hence, we are left with some obvious questions: what might be useful information to extract from books, how might we identify this information automatically, and how might we use this information fruitfully?

Our aim in this work is to move beyond using short pieces of text such as titles and abstracts, by using the full text of books (see also Betts *et. al.*, 2007) in order to identify useful information that can lead to improvements in browse and search capabilities, as well as recommendation strategies.

This research focuses on an attempt to identify the central characters in works of fiction, ordered by their importance in the story. We first hypothesise that the importance of characters is directly proportional to their frequency of mention. To measure the frequency accurately, we need to identify as many mentions of a character as possible and we speculate that this can only be predicted with the highest accuracy by first solving the problem of co-reference resolution.

Creating a Gold Standard

In order to test and evaluate our methods, we needed a gold standard – a set of books with collected relevant data. We chose 18 classic titles[1] from Project Gutenberg and designed a data collection experiment. We used a website[2] where participants could choose a title from a given list and fill in a form with the required information. Participants were asked to specify at least three and at most seven main characters (ordered by their importance to the story). We also provided space for specifying relationship information that we were using for a different part of the research (see Givon, 2006). We selected only the titles that had a significant amount of human data[3] and the information collected for the final 8 titles serves as our gold standard.

We measure the agreement between different users by computing agreement on the absolute order of characters. We first excluded statistically irrelevant data (e.g. reader responses that consisted of less than three characters). Agreement was computed between all possible pairs of readers where we counted the number of times when readers gave a character the same rank. This was divided by the maximum number of characters ranked by the readers (thus excluding characters that were not ranked by the pair of readers).

In agreement studies, 60% is considered adequate, and above 60% represents relatively good agreement (Landis and Koch, 1977). Our agreement test results, shown in Table 1, indicate that readers exhibit a high level of agreement on the top three central characters, and that agreement weakens as we add more central characters (down to 55.3% for the whole set). Hence, there is no significant agreement over all the voted characters.

	Top 1	**Top 2**	**Top 3**	**All**
Characters	97.8%	87.9%	68.4%	55.3%

Table 1: Absolute Order Agreement on Central Characters

When we came to compute ordered lists of characters to which we can compare our results, we found that our data contained some gaps, specifically where readers did not always rank every character in the superset. Therefore, to compute the order, we used the average score and number of voters. The formula attempted to account for both the average assigned ranks of the characters, and the number of voters, as we wanted the final rank to increase with fewer readers.[4] We computed a final rank rf_n for character n using the following formula:

$$rf_n = \frac{\sum_{i=1}^{k} r_{i,n} * j}{{k_n}^2}$$

[1] We selected the titles according to their sale records on Amazon.com.

[2] http://sgivon.tripod.com/Index.html

[3] The selected books are: Pride and Prejudice, Jane Ayre, Peter Pan, Little Women, Emma, Alice in Wonderland, Wuthering Heights and Great Expectations.

[4] Recall that a higher rank represents lower importance.

where $r_{i,n}$ is rank i for character n; k_n is the number of readers giving ranks to character n; and j is the total number of valid reader responses for the book.

Identifying Central Characters

All of our experiments used a set of existing pre-processing methods including tokenisation, sentence boundary detection, part of speech (POS) tagging (Curran and Clark, 2003), and chunking (Grover and Tobin, 2006). Given that we were only interested in the names of people, we used part of speech information to detect those names. We detected all the sequences of tokens tagged as proper nouns and maintained them in a list along with their original position in the text.

To identify co-occurrences, we used extracted attributes such as title, given names, surnames, and suffixes from our proper noun sequences. We used these attributes to compare names and identify whether they referred to the same character. This was done using a set of hand-crafted rules and an ensemble of gazetteers of titles, suffixes, given names, surnames and nicknames.

In many cases, referring to the same character using different names is very common due to the extensive use of prefixes[5]. We implemented a rule-based algorithm that resolves co-references[6]. E.g. a rule can be “Match characters with same prefix and last name”. To resolve ambiguous cases we experimented with two methods: (i) using location information (mainly closest proximity); and (ii) using highest term-frequency (tf) score (preferring a more frequent name to a less frequent one in the current text). At the end of this process, each name on the list was assigned an ID were names that refer to the same character were assigned the same ID. To automatically sort the list of characters by importance, we applied two frequency-based methods: (i) by a simple count of the number of mentions of names (based on allocated IDs); and (ii) by the tf-idf weight (Salton and Buckley, 1988)[7].

Results & Discussion

Our algorithms produced lists of central characters ordered by importance. To evaluate them, we used 10-best lists, and then compared these lists to our gold standard. We compared the lists in terms of absolute order and coverage. Based on this comparison, we computed an average score for our test sets. Finally, we computed precision, recall and F-score values. We evaluated our methods against a baseline state-of-the-art NER system.We present the results across various samples of the books, taking into consideration the unusual characteristics of some books (e.g. books written in the first person or books involving non-human characters).

[5] In Pride and Prejudice for example, the author refers to the main character, Elizabeth Bennet by eight different names: Elizabeth, Lizzy, Eliza, Miss Elizabeth, Miss Lizzy, Miss Eliza, Miss Elizabeth Bennet, and Miss Eliza Bennet.

[6] The algorithm is mainly designed to detect co-references between human entities (i.e. strings that represent names or titles of people) and it does not handle non-human entities.

[7] To compute our idf score we use a collection of 1,000 English and American Literature books from Project Gutenberg.

In terms of absolute order, we found that the best results were generated when we used the tf-idf score method with co-references by contextual information (location). This method achieved the best results on each sample of our test set. Additionally, the results improved in accuracy as we narrowed the character set and they were perfect on the top character. This suggests that more prominent characters are easier to extract.

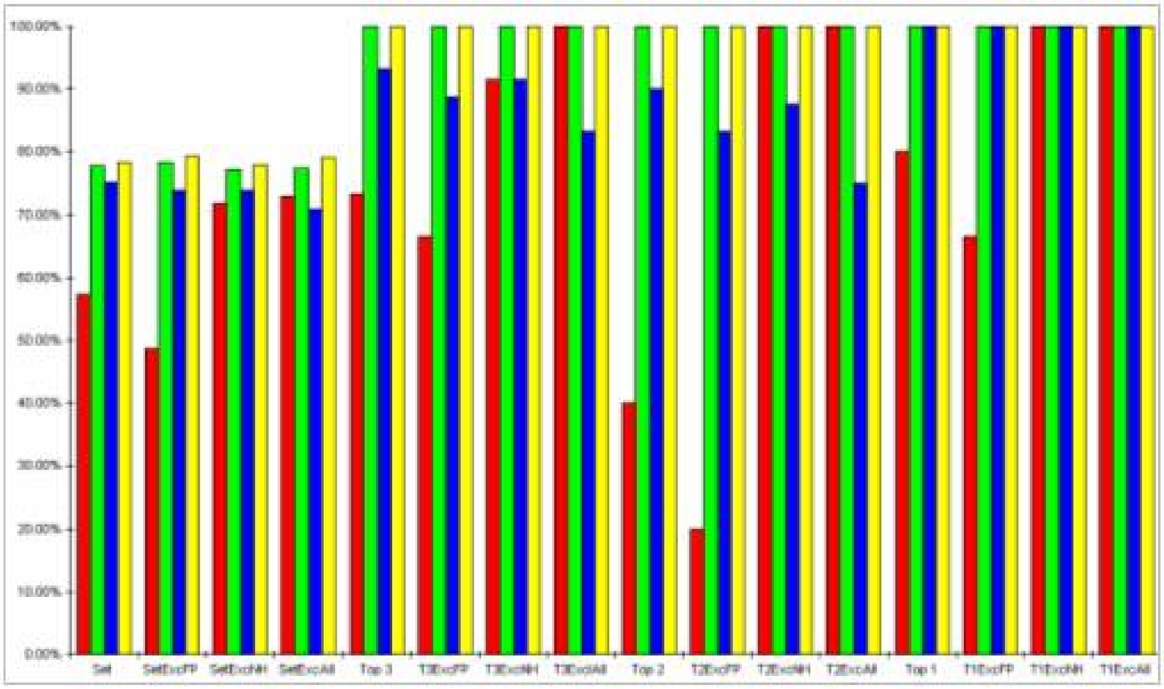

Figure 1: Coverage results where: T1 is top 1, T2 is top 2, T3 is top 3, FP is First Person, and NH is Non-Human. tfidf by tf is the tf-idf method with co-reference resolution by tf scores, and tfidf by context is the tf-idf method with co-reference resolution by contextual information

🟥 Baseline
🟦 tfidf by tf
🟩 tfidf by context
🟨 Frequency Count

Our coverage results were particularly good, as shown in Figure 1, and our baseline NER method yielded the worst results. The best results were produced by the frequency count algorithm, which resulted in 100% coverage for the top three characters in the book, and 78.4% across all characters. In terms of absolute order we obtained a best score of 100% for the most important character in the book, 62.5% for the top two characters and 49.9% for the top three. This matches the agreement level we showed earlier where significant agreement was only found on the top three or less characters. Table 2 shows the results for precision and recall on the whole set, where our best F-score of 64% was generated by the frequency count method when excluding all exceptions.

From the results, we can deduce that there is indeed a correlation between the frequency of the mentions of characters in a book and their importance. However, we cannot conclude that they are significantly proportional. To show this, methods that strongly affect absolute order (e.g. co-reference resolution) have to be improved. Our results clearly indicate that co-reference resolution is critical to identifying the central characters, particularly in books that involve more than one character with the same first or last name. Additionally, co-reference resolution strongly affects absolute order. In Pride and Prejudice, for example, we found 65 occurrences of 'Miss Bennet'. Without co-reference resolution we could not tell which Miss Bennet in particular each of them referred to (out of the possible five sisters). Moreover, in this book, absolute order accuracy increased from 33.3% on the top three characters and 10% on the whole set to 100% and 66.6% respectively after applying co-reference resolution to the data. Both methods that used co-reference resolution by contextual information obtained better results.

	Precision	**Recall**	**F-score**
Baseline	50%	72.5%	58.5%
tf-idf (1)[10]	50%	77.5%	59.5%
tf-idf (2)[11]	45%	64%	52%
Frequency	55%	79%	64%

Table 2: F-score results when excluding all exceptions

Although it is promising, the baseline NER algorithm obtained the worst results. Using the state-of-the-art NER system was our first choice for experimenting with detecting characters in works of fiction. However, in many aspects, this method was not ideal for the type of data used in this research (i.e. it was trained on newswire articles, which are quite different in style and length from fiction).

Conclusions

In this paper, we have investigated the scalability of IE techniques to full-sized books, and their utility in extracting useful information from fiction. Our analyses indicate that processing the full text of books is both feasible and useful. We have shown that it is possible to extract the central characters in the full text of a work of fiction relatively well, particularly in terms of coverage. We found that all of our methods performed better than our NER baseline method. However, in terms of absolute order we found that the NER method performed better on some samples of the set. This was mainly due to the more mature baseline algorithm that extracted name components.

We found that the importance of characters is directly proportional to their frequency in the text, which indicates that inferring the importance of characters in the text from frequency is acceptable. We also found that resolving co-references is critical for character extraction, particularly in terms of absolute order.

Future Work

Our initial investigations have not fully addressed all the issues of scalability when considering the full texts of books. We have identified that the tools[12] we were working with did scale to this task (a full book text is processed in approximately 5-10 minutes).

[10] tf-idf with co-reference resolution by contextual information.

[11] tf-idf with co-reference resolution by tf scores.

[12] Specifically, LT-XML (Thompson *et al.*, 1997), LT-TTT (Grover *et al.*, 2000) and LT-TTT2 (Grover & Tobin, 2006).

However, see Betts (2006) for further analyses of the performance of other existing systems on the full text of books, in which he failed to discover any other systems that could process the full text within a reasonable timeframe and produce useful results. We aim to investigate this issue further.

There are a number of areas where we can improve our character extraction algorithms. In particular, the tf-idf calculation needs to be enhanced to account for all the forms a name can occur in. We will also attempt to extend the types of extracted entity names we can deal with beyond only person names as well as add the ability to handle pronouns and non-human names.

Returning to the motivation behind this research, we see this work as a step towards creating a solution for automatic summarization and book comparison. However, to do this, more information should be extracted, such as relationships and main story events. The extracted information can also be beneficial for categorisation and aggregation purposes and can also enable the enhancement of recommendation methods.

Acknowledgements

We would like to thank Robert Japp and Mirella Lapata for their assistance and support throughout this research.

References

Betts, T. (2006). Using Text Mining to Place Books into an Ontology. M.Sc. Thesis. University of Edinburgh.

Betts, T., Milosavljevic, M. & Oberlander, J. (2007). The Utility of Information Extraction in the Classification of Books. In *Proceedings of the European Conference on Information Retrieval (ECIR'07)*. Rome, Italy, 2—5 April 2007.

Crane, G. (2006). What Do You Do with a Million Books? *D-Lib Magazine*. 12(3).

Curran, J. R., Clark S. (2003). Language Independent NER Using a Maximum Entropy Tagger. In *Proceedings of the 7th Conference on Natural Language Learning (CoNLL-03)*, 164-167, Edmonton, Canada.

Givon, S. (2006). Extracting Information from Fiction. M.Sc. Thesis. University of Edinburgh.

Grover C., Matheson, C., Mikheev, A., and Moens, M. (2000). LT TTT - a flexible tokenisation tool. In *Proceedings of LREC-2000*.

Grover, C., Tobin, R. (2006). Rule-Based Chunking and Reusability. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genoa, Italy.

Landis, J. R., Coch, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Journal of Biometrics* , 33, 159--174

Salton, G. & Buckley, C. (1988). Term Weighting Approaches in Automatic Text retrieval. *Journal of Information Processing and Management*, 24(5), 513--523.

Thompson, H., Tobin, R., McKelvie, D. and Brew, C. (1997). LT XML - software API and toolkit for XML processing. Downloadable from http://www.ltg.ed.ac.uk/software/.