Linking Task: Identifying authors and book titles in verbose queries

Anaïs Ollagnier, Sébastien Fournier, and Patrice Bellot

Aix-Marseille University, CNRS, ENSAM, University of Toulon, LSIS UMR 7296, 13397, Marseille, France.
Aix-Marseille University, CNRS, CLEO OpenEdition UMS 3287, 13451, Marseille, France.// {anais.ollagnier,sebastien.fournier,patrice.bellot}@univ-amu.fr

Abstract. In this paper, we present our contribution in INEX 2016 Social Book Search Track. This year, we participate in a new track called Mining track. This track focuses on detecting and linking book titles in online book discussion forums. We propose a supervised approach based on Support Vector Machine (SVM) classification process combined with Conditional Random Fields (CRF) to detect book titles. Then, we use a Levenshtein distance to link books to their unique book ID.

Keywords: Supervised approach, Support Vector Machine, Conditional Random Fields, References detection

1 Introduction

The Social Book Search (SBS) Tracks [2] was introduced by INEX in 2010 with the purpose of evaluate approaches for supporting users in searching collections of books based on book metadata and associated user-generated content. Since new issues have emerged. This year, a new track is proposed called Mining Track. This track includes two tasks: classification task and linking task. As part of our work, we focuse on the linking task, which consists to recognize book titles in posts and link them to their unique book ID. The goal is to identify which books are mentioned in posts. It is not necessary to identify the exact phrase that refers to book but to get the book that match the title in the collection.
The SBS task builds on a training corpus of topics which consists of a set of 200 threads (3619 posts) labeled with touchstones which allow can be used by members to easily identify books they mention in the topic thread, giving other readers of the thread direct access to a book record in LibraryThing[1] (LT), with associated ISBNs and links to collection. These posts are expressed in natural language made by users of LT forums. A data set contains book IDs, basic title and author metadata for each book. In addition, it is possible to use the document collection used in the Suggestion Track, which can be used as additional book metadata. This document collection consists of book descriptions for 2.8 million books.

[1] https://www.librarything.com/

In our contribution at SBS task, we use an approach inspired by the works on the bibliographical references detection in Scholarly Publications [1]. We propose a supervised approach based on classification process combined with Conditional Random Fields (CRF) to detect book titles. Then, we use the Levenshtein distance to link books to their unique book ID.
We submit 5 runs in which we compare several variations of selected features provide by CRF, of the combination of detected tags (book titles and author names) and of the factor taken by the Levenshtein distance.
The rest of this paper is organized as follows. The following section describes our approach. In section 3, we describe the submitted runs. We present the obtained results in section 4.

2 Supervised Approach for book detection

In this section, we present our supervised approach dedicated to book titles detection and link to their unique book ID. Firstly, we define a classification process with Support Vector Machine (SVM). Secondly, we describe the implementation of the CRF used. Thirdly, we present the use of Levenshtein distance to link books to their unique book ID.

2.1 Retrieving posts with book titles using SVM

In context of online forums, we have a wide variety of themes that are addressed. So, it is necessary to conduct a pre-filtering on verbose queries for identify queries with book titles. We choose to use a supervised classification based on SVM. We define two classes: "bibliographic field" versus "no bibliographic field". We establish a manual training set extracts randomly from the threads provided for the task. The class "bibliographic field" contains 184 posts and the class "no bibliographic field" 153 posts. For the SVM implementation, we use *SVMLight*[2] [4].
Regarding the settings, we use Weka[3] in order to establish a list of the most characteristic words of our classes, we use as attributes. Figure 1 shows an example of this list. The first column designates the score of "*Recursive Feature Elimination*" (RFE) and the second column refers to word. This list is obtained by

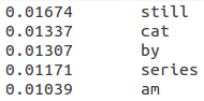

Fig. 1. Example of the most characteristic words of our classes

[2] http://svmlight.joachims.org/
[3] http://www.cs.waikato.ac.nz/ml/weka/downloading.html

the algorithm *InfoGainAttribute* which reduces a bias of multi-valued attributes. After several tests, we decided to use 1 as a minimum occurrence frequency of terms combined with a list from which we have removed the words with score RFE equal 0. This list is composed of 2182 words. We conduct 10-fold cross-validations to assess how the results generalize to a set of data. We obtain an accuracy of 80.69% with a recall of 100.00% and a precision of 78.45%.

2.2 Authors and book titles detection based on CRF

As part of our work, we choose to use an approach based on CRF. We establish a training set extracts randomly from the threads provide for the task. We manually annotated 133 posts in which we marked both book titles and author names. In total, we annotated 264 book titles and 203 author names. For constructing our CRF, we use several features presented in tables 1 and 2. For the CRF implementation, we use the tool *Wapiti*[4].
Main characteristics exploited in the literature for the automatic annotation of references are based on a number of observations such as lexical or morphological characteristics, both on the fields and the words contained in the fields. Drawing a parallel between the task of named entities detection and the analysis of bibliographic references, we are able to extract more useful information in the characterization fields and words contained in the fields. As part of our work, we decide to use a typology of features inspired by the literature.

- **Contextual features**: We add several features using the other tokens around the current one as three preceding and three following tokens. We use grammatical properties provides by a POS Tagger[5]. We use a natural language parser which provide structure of sentences. The latter two features, allow us to establish patterns of syntactic which define, independently words, potential anchors with greater portability. Table 1 describes these features.

Feature category	Description
Raw input token	Tokenized word itself in the input string and the lowercased word
Preceding or following tokens	Three preceding and three following tokens of current token
N-gram	Aggregation of preceding or following N-gram tokens
POS Tagger	Aggregation of preceding or following grammatical properties
NL Parser	Aggregation of preceding or following syntactical properties

Table 1. Description of contextual features

[4] `https://wapiti.limsi.fr/`
[5] `http://nlp.stanford.edu/software/tagger.html`

- **Local features**: they are divided into four categories: morphological, local, lexical and syntactic characteristics. The morphological features that were selected to characterize the shape of the tokens. The locational features have been selected to define the position of the fields in a sequence. The lexical features have been selected to use lists of predefined words but also linguistic category of words. And lastly, the punctuation features. Table 2 describes in contextual features.

Morphological features

Feature category	Feature name	Description
Number	ALLNUMBERS	All characters are numbers
	NUMBERS	One or more characters are numbers
	DASH	One or more dashes are included in numbers
Capitalization	ALLCAPS	All characters are capital letters
	FIRSTCAP	First character is capital letter
	ALLSAMLL	All characters are lower cased
	NONIMPCAP	Capital letters are mixed
Regular form	INITIAL	Initialized expression
	WEBLINK	Regular expression for web pages
Emphasis	ITALIC	Italic characters
Stem	-	Transformation in their radical or root
Lemma	-	Canonical form of current token form

Locational features

Location	BIBL_START	Position is in the first one-third of reference
	BIBL_IN	Position is between the one-third and two-third
	BIBL_END	Position is between the two-third and the end

Lexical features

Lexicon	POSSEDITOR	Possible for the abbreviation of editor
	POSSPAGE	Possible for the abbreviation of page
	POSSMONTH	Possible for month
	POSSBIBLSCOP	Possible for abbreviation of bibliographic extension
	POSSROLE	Possible for abbreviation of roles of entities
External list	SURNAMELIST	Found in an external surname list
	FORENAMELIST	Found in an external forename list
	PLACELIST	Found in an external place list
	JOURNALLIST	Found in an external journal list
POS Simple	Set tags	Harmonized Part of speech

POS Detail	Set tags	Detailed Part of speech
Punctuation features		
Punctuation	COMMA	Punctuation type.
	POINT	
	LINK	
	PUNC	
	LEADINGQUOTES	
	ENDINGQUOTES	
	PAIREDBRACES	

Table 2: Description of local features

From these characteristics we construct vectors for each word.
Following the classification, we get a list of posts containing book titles potentially. Then, our CRF allows us to annotate the area referring to book titles or author names.

2.3 Mapping to book Ids

Once book titles or author names detection carried out, we use the Levenshtein distance for link books to their unique book ID. Just for recall, Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits required to change one word into the other. As part of our work, we use two variations of the Levenshtein distance: either the length of the shortest alignment between the sequences is taken as factor, or the length of the longer one.
For each book title found, we compare it with the whole of book titles extracts from the collection. For each book title, we obtained a list of the books sorted by normalized Levenshtein distance, so that the results of several distance measures can be meaningfully compared. Figure 2 presents the best three results obtained for the book entitled "The Old Man".

```
Book Name: The Old Man
Results: (' The plain old man', 0.75) (' The happy old man', 0.75) (' The Old Man', 1.0)
```

Fig. 2. Example output for the book title "The Old Man"

For each book title, we keep the best result which is close to 1. Then, we retrieve the unique ID of the most probable best book. If an author name is located at a maximum distance of four words, we aggregate it. Figure 3 shows a query which contains both book title and author name. Then, figure presents the result obtained for the input "Timothy Findley The Last of the Crazy People".

```
   <message>

<date>Dec 8, 2010, 11:44am </date>
<note>I'm currently reading <book>The Town That Forgot How to Breathe</book>
by <author>Kenneth J. Harvey</author>. The back cover describes it as a gothic
 thriller and it's set in Newfoundland. So far, about 150 pages into it,
it's been building an increasingly creepy atmosphere.</note>
<postid>7</postid>

<threadid>103956</threadid>

<username>TheTwoDs</username>
   </message>
```

Fig. 3. Example of post with book title and author name

```
Author + Book :  Timothy Findley   The Last of the Crazy People

Result (' The Last of the Crazy People ', 0.9)
```

Fig. 4. Result obtained for the input "Timothy Findley The Last of the Crazy People"

3 Runs

We submitted 5 runs for the linking task of Mining track. For each run, we use only the data set which contains book IDs, basic title and author metadata per book. Once the classification process and the annotation process done, we link books at the post level by their unique LibraryThing work ID. Concerning a book which occurs multiple times in the same post, we keep only the first occurrence. Figure 5 shows an example of the second post of the thread 16512. For each post, we have the content of the post, the name of the user, the thread id as well as the date and time.

```
  <message>
<note>I am a school librarian and these are loved by boys this age.....  <book>Time
Warp trio</book> by <author>Scieszka – 3</author> boys go back in time and have
all sorts of adventures <book>Boxcar Children</book> by Warner are great
mysteries <book>Geronimno Stilton</book> by Stilton are more action mysteries
told by a mouse <book>Hank the Cowdog</book> by Ericson funny adventures
Good Luck! </note>
<postid>2</postid>

<username>KC9333</username>

<threadid>16512</threadid>

<date>Jul 18, 2007, 10:01pm </date>
  </message>
```

Fig. 5. Example of post for the thread 16512

Figure 6 shows the result obtained for this post. The first column corresponds to the thread id. The second column defines the post id. The third column returns

the unique LibraryThing work ID (what is shown in brackets is not present in the final version of the results file.).

```
16512   2   94411       (The Time Warp Trio)
16512   2   9397335     (The Boxcar Children)
16512   2   7487288     (Geronimo Stilton Adventurer's Boxed Set)
16512   2   67423       (Hank the Cowdog)
```

Fig. 6. Example of results for the second post of the thread 16512

Let's now explain the different runs:

- **B:**
  After the classification process and the annotation process, we retrieve each book title and we compare it with the whole of the titles presents within the data set. This comparison is carried out by the Levenshtein distance set to the length of the shortest alignment between the sequences taken as factor.
- **B_V2:**
  After the classification process and the annotation process, we retrieve each book title and we compare it with the whole of the titles presents within the data set. This comparison is carried out by the Levenshtein distance set to the length of the longer alignment between the sequences taken as factor.
- **BU:**
  For this run, we add a new feature to the CRF. This feature is to detail the punctuation marks. Once the classification process and the annotation process are done, we retrieve each book title and we compare it with the whole of the titles presents within the data set. This comparison is carried out by the Levenshtein distance set to the length of the shortest alignment between the sequences taken as factor.
- **BA_V1**
  After the classification process and the annotation process, we retrieve each book title. If an author name is located at a maximum distance of four words, we aggregate it. Then, we compare the book title and the author name, if it is present, with the information present within the data set. This comparison is carried out by the Levenshtein distance set to the length of the shortest alignment between the sequences taken as factor.
- **BA_V2:**
  After the classification process and the annotation process, we retrieve each book title. If an author name is located at a maximum distance of four words, we aggregate it. Then, we compare the book title and the author name, if it is present, with the information presents within the data set. This comparison is carried out by the Levenshtein distance set to the length of the longer alignment between the sequences taken as factor.

4 Results

For evaluation, 217 threads in the test set were used, with 5097 book titles identified in 2117 posts. Table 3 shows 2016 official results for our 5 runs. Our best run is BA_V2, it has classified the second w.r.t the measure Fscore and the first w.r.t the measure precision the official evaluation measure for the workshop. The others runs have substantially similar results. However, we can see that the aggregation of author names increases performance. Compared to the best run 2016, the whole of our runs get a better precision. Several hypotheses may explain the lack of recall. Firstly, the classification process can occult posts containing references. Secondly, the amount of training data may not be enough to be representative of every possible case.

Run	Accuracy	Recall	Precision	Fscore
Best_run_2016	41.14	41.14	28.26	33.50
BA_V2	**26.99**	**26.99**	**38.23**	**31.64**
BA_V1	26.54	26.54	37.58	31.11
B_V2	26.01	26.01	35.39	29.98
BU	26.34	26.34	34.50	29.87
B	25.54	25.54	34.80	29.46

Table 3. Official result at INEX 2016. The runs are ranked according to Fscore

5 Conclusion

In this paper we presented our contribution for the INEX 2016 Social Book Search Track. In the 5 submitted runs, we tested several supervised approaches dedicated to book detection. Our results present better performance with the aggregation of author names. Moreover, the Levenshtein distance set to the length of the longer alignment between the sequences taken as factor give better results than the shortest alignment.

References

1. Ollagnier A., Fournier S., Bellot P.: A supervised Approach for detecting allusive bibliographical references in scholarly publications. In: 6th WIMS Web-Intelligence, Mining and Semantics. (2016)
2. Kazai, G., Koolen, M., Kamps, J., Doucet, A., Landoni, M.: Overview of the INEX 2010 book track: Scaling up the evaluation using crowdsourcing. In: Comparative Evaluation of Focused Retrieval. pp. 98–117. (2010)
3. Councill, I., Giles, C., Kan, M.-Y.: ParsCit: An open-source CRF reference string parsing package In: LREC. European Language Resources Association. (2008)
4. Joachims T.: Optimizing Search Engines Using Clickthrough Data. In: Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD). (2002).
5. Ren J.: ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging. Knowledge-Based Systems 26. pp. 144–153 (2012).