2012 International Conference on Networks and Information (ICNI 2012)
IPCSIT vol. 57 (2012) © (2012) IACSIT Press, Singapore
DOI: 10.7763/IPCSIT.2012.V57.15

Role of Apostrophes in Turkish Information Retrieval

Ahmet Arslan [1+] and Ozgur Yilmazel

[1] Computer Engineering Department, Anadolu University Eskisehir, Turkey

Abstract. Unlike structured information access, unstructured information access or text retrieval is language dependent. In this work we discuss how apostrophes should be handled in Turkish text retrieval. We conduct experiments over a TREC-like dataset for Turkish using Apache Solr 4.0-BETA as retrieval engine. Our experiments showed that our proposed way of handling apostrophes significantly increased retrieval quality for Turkish language.

Keywords: apostrophe, lucene, solr, retrieval, Turkish.

1. Introduction

Apache Solr is the popular, blazing fast open source enterprise search platform from the Apache Lucene project. Solr is built on top of Lucene, which is an open source, high-performance text search engine library. They are both open source and written entirely in Java. The Lucene community has recently decided to merge the development of Lucene and Solr. Both code bases now sit under the same trunk in subversion and Solr consumes the latest Lucene code at all times.

Text analysis, in Solr/Lucene, is the process where plain text is converted into a stream of tokens by tokenization, lowercasing, stemming, synonym expansion, stopword removal and etc.

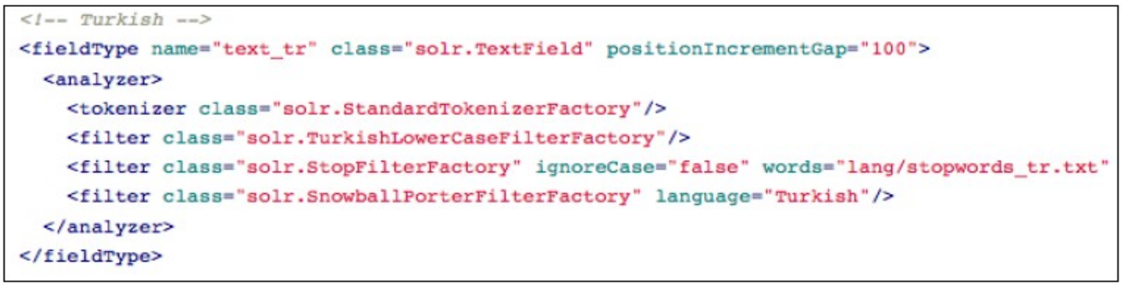
"Analysis chains consist of character filters (useful for stripping diacritics, for instance), tokenizers (which are the sources of token streams) and series of token filters that modify the original token stream. Lucene includes a total of five character filtering implementations, 18 tokenization strategies and 97 token filtering implementations and covers 32 different languages" [1].

Solr's field type definition allows creating complex analysis pipes using these existing character filters, tokenizers and token filters. In a field type's analyzer section there can be zero or more character filters, only one tokenizer, and zero or more token filters. Solr's example schema.xml contains configurations for various languages as field types (text_tr, text_ru, text_fr, etc). Fig. 1 shows field type definition (text_tr) created for Turkish language. It has Turkish-specific lowercasing, stemming and stopword removal. Solr includes example set of Turkish stopwords that can be found in `lang/stopwords_tr.txt` file.

Solr has nice User Interface that visualizes each analysis step. Fig. 2 shows analysis chain steps when sample text (which is description part of topic number 352) is analysed using text_tr field type. Note that Analysis User Interface uses acronyms when displaying names of components.

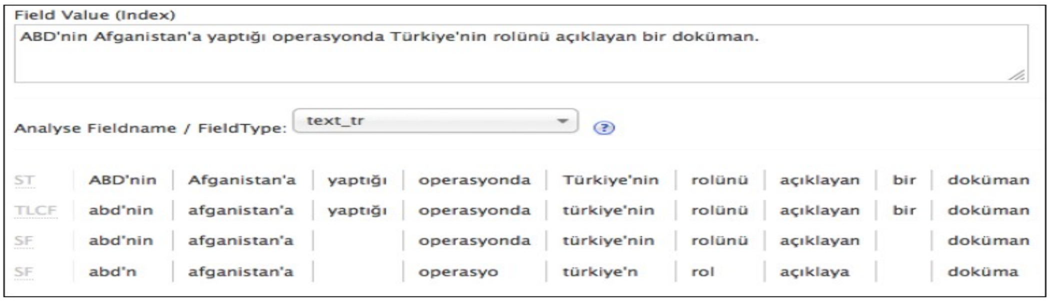
- StandardTokenizer (ST): Responsible for breaking up incoming text into tokens.
- TurkishLowerCaseFilter (TLCF): Lowercases token text. (takes account into dotted and dotless I)
- StopFilter (SF): Removes stopwords from token streams. ("yaptığı" and "bir" are removed)
- SnowballFilter (SF): Stems words using a Snowball-generated stemmer. ("rolünü" → "rol")

[+] Corresponding author. Tel.: + 90 532 620 3599; fax: +90 (222) 323 95 01.
E-mail address: aarslan2@anadolu.edu.tr



```
<!-- Turkish -->
<fieldType name="text_tr" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.TurkishLowerCaseFilterFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="false" words="lang/stopwords_tr.txt"
    <filter class="solr.SnowballPorterFilterFactory" language="Turkish"/>
  </analyzer>
</fieldType>
```

Fig. 1: text_tr field type analysis chain components.



Field Value (Index)

ABD'nin Afganistan'a yaptığı operasyonda Türkiye'nin rolünü açıklayan bir doküman.

Analyse Fieldname / FieldType: text_tr

ST	ABD'nin	Afganistan'a	yaptığı	operasyonda	Türkiye'nin	rolünü	açıklayan	bir	doküman
TLCF	abd'nin	afganistan'a	yaptığı	operasyonda	türkiye'nin	rolünü	açıklayan	bir	doküman
SF	abd'nin	afganistan'a		operasyonda	türkiye'nin	rolünü	açıklayan		doküman
SF	abd'n	afganistan'a		operasyo	türkiye'n	rol	açıklaya		doküma

Fig. 2: text_tr field type sample text analysis.

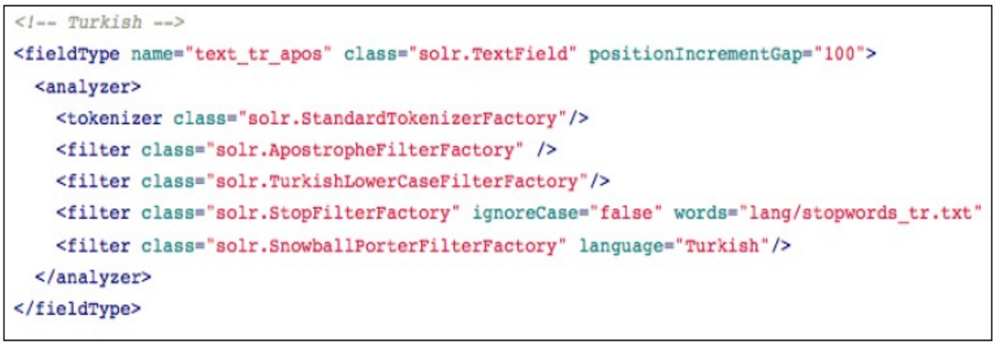
In our previous [2] work we compared different stemming algorithms using Lucene and used pretty much same analysis chain as in text_tr field type for snowball stemmer. Only difference was we had StandardFilter at that time. We improved our Turkish Language Analysis as explained in this work.

With text_tr field type, the query string türkiye won't retrieve the sample text in Fig. 2 because the word Türkiye'nin is indexed as türkiye'n. To address this issue we have implemented a custom ApostropheFilter that removes all characters after an apostrophe (including the apostrophe itself). In Turkish, apostrophe is used to separate suffixes from proper names (continent, sea, river, lake, mountain, upland, proper names related to religion and mythology). For example Van Gölü'ne (meaning: to Lake Van).

Although it seems removing apostrophe and remaining character is similar to stemming, it must be done by a separate filter and before the stemming. Proper names usually are in the root or stem form already. But there are some cases where proper names are inflected. For example the input string Van Gölü'ne will be reduced to Van Gölü after apostrophe filter. And then it will be reduced to Van Göl after the stemming so that query göl (meaning: lake) will match that input string. Also in Turkish, more than one apostrophe does not occur in single word. But this can happen in English. e.g. O'Reilly's.

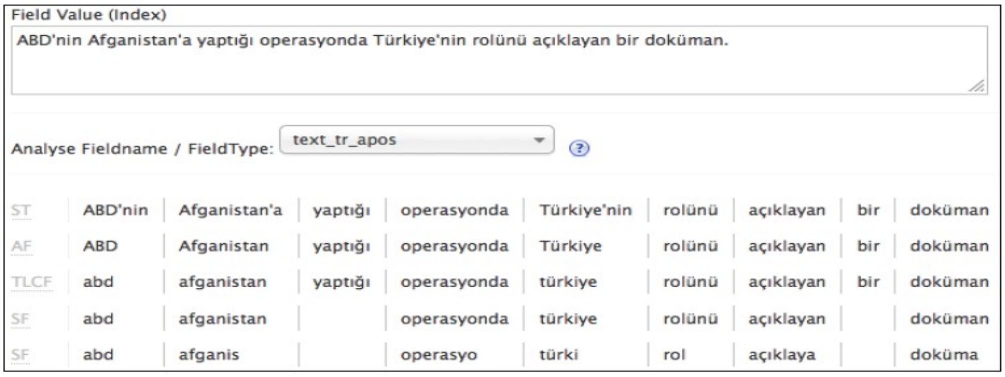
Solr/Lucene has a similar built-in token filter named StandardFilter that removes 's from the end of words. But this is useful for English language only and has nothing to do with Turkish. Because it only removes 's and 'S. In Turkish different character sequences comes after an apostrophe. e.g. 'nin, 'a, 'nin, 'ü etc.

We place ApostropheFilter right under the standard tokenizer. Fig. 3 shows our modified field type configuration that we call text_tr_apos. Fig. 4 shows analysis chain steps of text_tr_apos field type when the same sample text used in Fig. 2. Note that indexed tokens are different than Fig. 2. The word Türkiye'nin is indexed as türki. Query türkiye will now retrieve the sample text with this setting. Note that, query string türkiye will be reduced to türki at query time by SnowballFilter (stemmer).



```
<!-- Turkish -->
<fieldType name="text_tr_apos" class="solr.TextField" positionIncrementGap="100">
  <analyzer>
    <tokenizer class="solr.StandardTokenizerFactory"/>
    <filter class="solr.ApostropheFilterFactory" />
    <filter class="solr.TurkishLowerCaseFilterFactory"/>
    <filter class="solr.StopFilterFactory" ignoreCase="false" words="lang/stopwords_tr.txt"
    <filter class="solr.SnowballPorterFilterFactory" language="Turkish"/>
  </analyzer>
</fieldType>
```

Fig. 3: text_tr_apos field type analysis chain components.



Field Value (Index)

ABD'nin Afganistan'a yaptığı operasyonda Türkiye'nin rolünü açıklayan bir doküman.

Analyse Fieldname / FieldType: text_tr_apos

ST	ABD'nin	Afganistan'a	yaptığı	operasyonda	Türkiye'nin	rolünü	açıklayan	bir	doküman
AF	ABD	Afganistan	yaptığı	operasyonda	Türkiye	rolünü	açıklayan	bir	doküman
TLCF	abd	afganistan	yaptığı	operasyonda	türkiye	rolünü	açıklayan	bir	doküman
SF	abd	afganistan		operasyonda	türkiye	rolünü	açıklayan		doküman
SF	abd	afganis		operasyo	türki	rol	açıklaya		doküma

Fig. 4: text_tr_apos field type sample text analysis.

2. Dataset

Information Retrieval studies mostly focus on English language and there are Information Retrieval Evaluation Forums (such as TREC[2] and INEX[3]) for English language. These Information Rretrieval Forums create test collections and provide standardized evaluation of search engines. Unfortunately there are no such events, annual competitions for Turkish language text retrieval.

In 2008, Fazli Can and Bilkent Information Retrieval Group created (Milliyet Collection) the first large-scale Turkish Information Retrieval test collection and made this TREC-like test collection available to the other researchers [3]. The collection consists of 408305 documents (news stories), 72 information needs (called topics in TREC) and relevance judgments (right answers); the collection is roughly 800MB in size. In this paper we used this dataset because it is the only one reusable, large-scale dataset for Turkish langue.

3. Experimental Setup and Results

Each document in Milliyet Collection consists of eight fields: {author, date, DOCNO, headline, source, text, time, URL}. Among these eight fields only headline and text fields contain searchable textual information. Therefore we combined text and headline fields using Solr's copy field declaration as follows:

<copyField source="headline" dest="content_tr"/> <copyField source="headline" dest="content_tr_apos"/>

<copyField source="text" dest="content_tr"/> <copyField source="text" dest="content_tr_apos"/>

Thus we indexed Milliyet Collection documents into two different fields (content_tr and content_tr_apos) using text_tr and text_tr_apos field types respectively.

Information needs in Milliyet Collection consists of three parts (title, description and narrative), which are similar to a typical TREC query. We didn't develop any special query construction technique to obtain queries from topics. We used title and description portions of the topics as it is and ignored narrative portion. Only modification we made to title and description parts is to strip question mark (?) from them. Because in Solr/Lucene query parser syntax, **?** is a wildcard search operator.

Title-only queries and title&description queries are fired over content_tr and content_tr_apos fields in our runs. Maximum 1000 documents (per topic) are fetched from Solr, similar to standard TREC-type ad hoc runs. Tab.1 show several evaluation metrics produced by trec_eval[4] utility (version 8.1), which is the standard tool, used by the TREC community for evaluating an ad hoc retrieval run.

Having text_tr field type as baseline, text_tr_apos field type increased all evaluation measures in all runs. Interpolated average precision (MAP) increased 5.2%, precision at 5 (P@5) increased 2.9% and precision at 10 (P@10) increased %4.8 for title&description queries. Also number of unique terms decreased from 763092 to 621702 when collection is indexed using text_tr_apos field type.

[2] http://trec.nist.gov/

[3] https://inex.mmci.uni-saarland.de/

[4] http://trec.nist.gov/trec_eval/

Tab.1: Evaluation metric from title-only queries and title&description queries.

	title-only queries		title&description queries	
	text_tr	text_tr_apos	text_tr	text_tr_apos
num_rel_ret	5585	5769	6306	6493
map	0.3159	0.3298	0.3949	0.4154
gm_ap	0.2142	0.2309	0.3202	0.3483
R-prec	0.3518	0.3606	0.4158	0.4293
bpref	0.4318	0.4427	0.4861	0.5063
P@5	0.5528	0.5778	0.6806	0.7000
P@10	0.5472	0.5556	0.6625	0.6944

4. Conclusion

In this paper Milliyet Collection's ad hoc task is completed. Apache Solr's default language analysis configuration for Turkish is used as baseline. We showed that addition of custom ApostropheFilter (that strips apostrophe separated suffixes) to baseline configuration shrinks the number of unique terms and significantly increases retrieval quality.

Text retrieval is language dependent and every language has its own characteristics. For a retrieval engine to work effectively, all properties must be taken into account for that language. In this work we showed very simple and small language specific modification made to text analysis phase can dramatically affect retrieval performance.

5. References

[1] A. Białecki, R. Muir, G. Ingersoll. Apache Lucene 4. *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval*. Portland, Oregon, USA, 16th August 2012, pp. 17-24.

[2] A. Arslan, O. Yilmazel. A comparison of Relational Databases and information retrieval libraries on Turkish text retrieval. *Natural Language Processing and Knowledge Engineering Conference*, pp. 1-8, 19-22 Oct. 2008.

[3] F. Can, S. Kocberber, E. Balcik, C. Kaynak, H. C. Ocalan, O. M. Vursavas. Information retrieval on Turkish texts. *Journal of the American Society for Information Science and Technology*, 59(3): 407-421, 2008.