Available online at www.sciencedirect.com

ScienceDirect

Computer Speech & Language 56 (2019) 95–106

www.elsevier.com/locate/csl

Empirical evaluation of compounds indexing for Turkish texts☆

Chedi Bechikh Ali[a,*], Hatem Haddad[b], Yahya Slimani[c]

[a] LISI, INSAT, Universié de Carthage, Centre Urbain Nord BP 676, Tunis 1080, Tunisia
[b] Université Libre de Bruxelles, Avenue Franklin Roosevelt 50, Bruxelles 1050, Belgium
[c] Computer Science Department, ISAMM Institute of Manouba, Oued Ellil, Tunisia

Received 19 April 2017; received in revised form 21 February 2018; accepted 25 January 2019
Available online 5 February 2019

Abstract

In this article, we describe an empirical evaluation of compounds indexing for Turkish texts. We dive beyond the keyword indexing to propose a framework for Turkish compounds extraction and indexing. We identify twelve Turkish compounds pattern types that we classify in six categories. To extract Turkish compounds, we rely on a light natural language processing approach based on syntactic pattern recognition. We compare different compounds indexing strategies. We also investigate the effectiveness of using one compounds type and the effectiveness of combining different compound types. We conduct experiments over the Milliyet test dataset. The results of our experiments show that using compounds as index terms can improve retrieval performances. However, not all the compound types have a positive impact on the retrieval process.

Keywords: Information retrieval; Indexing; Turkish language; Compound; Syntactic patterns

1. Introduction

An information retrieval system (IRS) faces the challenge of retrieving and ordering relevant documents. Most information retrieval (IR) models use keywords as “bag of words”. Thus, documents (as well as queries) are represented as an unordered set of words. This word-based representation of documents and queries content is based on statistical weighting formula and the assumption of words independence. Earlier works proved that the use of “bag of words” as keywords is not accurate enough to represent textual contents due to the words ambiguity. Indeed, words are not specific enough for accurate discrimination (Sanderson, 1994), especially in the case of technical documents where compounds are very frequent (Tong et al., 1996; Arampatzis et al., 1998; Haddad, 2003). By compounds, we refer to complex terms and noun phrases. Preslav (2013) gives the different definitions of compounds used in the state-of-the-art. Many researchers enumerated “bag of words” representation limitations in the context of IR (Bendersky and Croft, 2008; Park et al., 2011; Preslav, 2013).

To achieve the highest precision, an IR system has to consider the use of a precise document (as well as query) content representation, different than the “bag of words” representation (Metzler and Croft, 2005; Lioma et al.,

☆This paper has been recommended for acceptance by Roger K. Moore.

* Corresponding author.
E-mail addresses: chedi.bechikh@gmail.com (C. Bechikh Ali), hatem.haddad@ulb.ac.be (H. Haddad), yahya.slimani@fst.rnu.tn (Y. Slimani).

https://doi.org/10.1016/j.csl.2019.01.004
0885-2308/ 2019 Elsevier Ltd. All rights reserved.

2015). For example, "*milli güvenlik*" (National security) is an important compound in the Bilkent Information Retrieval Group Test dataset called Milliyet (Can et al., 2006; 2008). The dataset contains 408,305 news articles and columns of five years, 2001−2005, from the Turkish newspaper Milliyet. Neither *milli* (National) nor *güvenlik* (Security) are important by themselves because they have high frequencies and minimal discriminating values. On the other hand, the "*milli güvenlik*" represents a specific concept in the dataset. By separating the two words in the indexing phase, the concept can not be represented. Using the words *milli* and *güvenlik* as a query can lead to an important retrieval documents number including documents not containing the compound "*milli güvenlik*".

A solution to this problem is to use compounds instead of words to index documents and queries. The assumption is that compounds are more likely to identify semantic entities than words, to make a better discriminator (Strzalkowski et al., 1999) and to better represent the semantic document (as well as the semantic query) content (Mitra et al., 1997; Bendersky et al., 2011; Gupta and Bendersky, 2015). We adopt the same definition of a compound as in Preslav (2013): *"long sequence of nouns that act as a single noun"*. Accordingly, we endorse the same compounding definition of (Trask, 1993): *"the process of forming a word by combining two or more existing words"*.

Earlier works focused on statistical techniques to extract compounds (Fagan, 1987; Buckley et al., 1995). Recent works propose linguistic techniques for the recognition and the extraction of compounds (Pickens and Croft, 2000; Bechikh Ali et al., 2015). Light natural language processing showed that it can improve matching results by combining classical IR methods with compounds recognition (Tong et al., 1996; Arampatzis et al., 2000; Haddad, 2003; Zhang et al., 2007).

In this article, we focus on Turkish compounds for IR. Due to the Turkish language characteristics, Turkish IR presents a significant challenge for the IR research community. Indeed, Turkish is an agglutinative language where words are constructed using inflectional and derivational suffixes linked to a root. For example, there are approximately 23,000 stems and 350-400 roots actively used. If we include inflection of the words, the number raises to millions and can be over a thousand forms of one verb (Sever and Bitirim, 2003). Because of the complexity of Turkish morphology, Turkish language needs a special attention when it comes to text processing and computational linguistics. For this reason, studies on Turkish IR focused on Turkish natural language processing rather than Turkish IR questions (Oflazer, 2014). Compared to other languages, Turkish did not have enough attention by the IR community. One main reason of that is the absence of a Turkish standard test dataset, except the Milliyet test dataset which is an incomplete dataset (Can et al., 2008).

Within the context of Turkish IR, we investigate the *Phrase Retrieval Hypothesis* stated in Arampatzis et al. (1998) and in Arampatzis et al. (2000). We dive beyond the keyword document and query indexing to use compounds as index. We hypothesis that the use of compounds to index Turkish documents and queries would improve IR performances, especially precision performances at low recall levels. To extract Turkish compounds, we rely on a light natural language processing approach based on syntactic pattern recognition (Strzalkowski et al., 1999). On this study, we focus on two-word compounds. We define twelve types of hand-crafted regular expression patterns. The proposed model is evaluated by using one compound type and by combining different compound types. We compare our proposed approach performances with the keyword indexing performances and with the state-of-the-art performances.

The rest of the article is organized as follows. Section 2 presents Turkish language characteristics as well as Turkish compounds. Section 3 outlines existing works on Turkish information retrieval. Section 4 describes our methodology to extract Turkish compounds. In section 5, we describe our experimental setup including the information retrieval system used, the test dataset, the text preprocessing and the retrieval evaluation metrics. Section 6, presents the results and discussion. Finally, limitations and future work are covered in Section 7.

2. Turkish language and Turkish compounds

In terms of number of speakers, Turkic language family is one of the largest languages in the World. In Can et al. (2008) and in Kornfilt (1997), the authors summarized the Turkish language history, alphabet and Turkish language grammar. The Turkish alphabet is based on Latin characters and has 29 letters consisting of 8 vowels and 21 consonants. It has all the letters of the English alphabet, except q, w, and x. In addition, it has 5 extra letters derived from Latin by adding diacritics: ç, ğ, ş, ü, ö, and ı (Arslan, 2016). Alpkocak and Ceylan (2012) compared the usage of these letters and the usage of these letters without their umlauts, dots or cedilla. Their IR system performances showed that using documents and queries with proper Turkish diacritics performs better than replacing them with standard Latin alphabet letters. In our study, we opt for this solution.

As in all languages, compounds have to have at least two unbounded morphemes (words) in Turkish (Göksel and Haznedar, 2007). The compounding process can be extremely productive in the Turkish language (Sever and Bitirim, 2003). We identify twelve Turkish compound pattern types that we classify into six categories (Kornfilt, 1997; Göknel, 2010):

(i) < Noun Noun > Compounds. This type is the most common in Turkish language. The compounds exist in three different types:

  (1) Pattern1: in this first type, the first noun has no inflectional suffix and the second one has to have a third person possessive suffix. It is called *Undefined Compound*; for examples: *okuma salon**u*** (reading hall), *duvar kağı**dı*** (wall paper).
  (2) Pattern2: in this compounds type, the first noun has genitive case -(n)In and the second noun has a third person possessive suffix. This pattern is named as *Defined Compound*; for examples: *kadı**nın** elbise**si*** (the dress of the woman), *kapı**nın** kol**u*** (the handle of the door).

(ii) < Adjective Noun > Compounds. There are three different compound types:

  (1) Pattern3: in this pattern, the adjective has no suffix; for examples: *büyük baba* (grand father), *kötü insan* (bad man), *beş evler* (five houses).
  (2) Pattern4: in this pattern, the adjective has a suffix deriving adjective from noun such as -lI or sIz; for examples: *tuz**lu** çorba* (salty soup), *şeker**siz** çay* (sugar free tea).
  (3) Pattern5: in this pattern, the adjective has a suffix deriving adjective from verb such as -(y)An, -mIş, -(ı)k, -(y)ecek, -(y)IcI, etc.; for examples: *koş**an** adam* (running man), *kır**ık** cam* (broken glass), *geç**miş** gün* (passed day), *gel**ecek** yıl* (comming year), *aldat**ıcı** düşünce* (misleading idea).

(iii) < Noun Adjective > Compounds. They exist in four different types

  (1) Pattern6: in the pattern type, a noun and an adjective are joint without having a derivatinal suffix; for examples: *kar beyaz* (snow white), *süt beyaz* (milk white).
  (2) Pattern7: in this pattern, the noun has to have a third person singular possessive suffix; for examples: *el**i**açık* (generous), *gönl**ü** zengin* (warmhearted).
  (3) Pattern8: in this pattern, the adjective is derived from a noun; for examples: *mavi göz-**lü*** (blue eyed), *cin fikir**li*** (ingenious).
  (4) Pattern9: in this pattern, the adjective is derived from a verb; for examples: *vatan sev**er*** (homeland lover), *bilgi say**ar*** (computer).

(iv) Pattern10: < Adjective Adjective > Compounds; for example: *aç gözlü* (hungry eye-with).
(v) Pattern11: < Verb Verb > Compounds: in this type, the compound is formed by two verbs; for example: *kaptı kaçtı* (snatch+past-flee+past).
(vi) Pattern12: < Noun Verb > Compounds; for examples: *gece kondu* (night-to be put+past “slum”), *hünkar beğendi* (king-like+past “a kind of dish”).

We present in Table 1 a list of the syntactic patterns that we used to extract compounds from documents and queries.

3. Turkish information retrieval

In this section, we survey previous studies on Turkish IR. Due to the lack of standard Turkish IR test datasets, Turkish IR is a field that has not achieved much interest compared to other languages (Can et al., 2008). Most of the studies in Turkish IR have built their own test datasets. Thereby, this makes it difficult to compare different approaches and their performances.

According to Can et al. (2008), the earliest published Turkish IR study is done by Köksal (1981) using 570 documents on computer science and 12 queries. The study evaluated the effectiveness of Turkish words truncation. After

Table 1
Used compound patterns.

Pattern	Description
Pattern1	Noun (with no suffix) Noun (ends with u, ü, i and ı)
Pattern2	Noun (ends with un, ün, in and ın) Noun (ends with u, ü, i and ı)
Pattern3	Adjective (no suffix) Noun
Pattern4	Adjective (ends with lu, lü, li, lı,suz, süz, sız and siz) Noun
Pattern5	Adjective (ends with an, en, mls, ecek, (ı)k, IcI) Noun
Pattern6	Noun Adjective (no suffix)
Pattern7	Noun (end with suffix) Adjective
Pattern8	Noun Adjective (ends with lu, lü, li, lı)
Pattern9	Noun Adjective (ends with er, ar)
Pattern10	Adjective Adjective
Pattern11	Verb Verb
Pattern12	Noun Verb

evaluating different prefix sizes, the author concluded that the use of the first 5 characters (5-prefix) of words is the best truncation approach. Solak and Can (1994) applied a stemmer in a dataset of 533 news articles obtained from a Turkish news agency. They used a morphological analysis processing a word in three steps: root determination, morphophonemic checks and morphological parsing. Authors concluded that stemming reduces the index dictionary size by 65%.

Sever and Bitirim (2003) proposed a stemming algorithm called FINDSTEM. The stemmer was evaluated using 2468 law documents and 15 queries in a vector space model. The authors concluded that stemming increases search precision by approximately 25% of the average precision values at 11-point recall levels when compared to no stemming.

Ekmekçioglu and Willett (2000) studied the effectiveness of queries stemming on Turkish IR. They used a Turkish dataset containing the titles and abstracts of 6289 economic and political news stories extracted from Turkish newspapers from 1991 to 1993. They used 50 queries prepared by 30 Turkish natives judges providing both natural-language queries and relevance judgments on the stemmed and unstemmed search outputs. In this experiment, the authors compared the retrieval effectiveness using stemmed and unstemmed query words (document words were used as they are). Stemming was achieved using the two-level morphological analyzer PC-KIMMO. Since different roots can correspond to one query word, the query was extended with all possible roots that can be related to the query words. Based on the OKAPI text retrieval system, the stemmed queries provided a number of relevant documents about 32% more than the unstemmed queries at the retrieval cut-off of 10 and 20 documents.

The first IR usage of the Milliyet dataset was presented in Can et al. (2008). The results of four different stemming options on Turkish IR effectiveness were compared: no stemming, simple word truncation, the successor variety method adapted to Turkish (Hafer and Weiss, 1974) and a lemmatizer-based stemmer for Turkish (Aysın and Fazlı, 1994). The authors also studied the effectiveness of using a stopword list. They concluded that, unlike other languages, in the case of the Turkish language a stopword list has no leverage on the IR system effectiveness. In addition to that, a simple word truncation approach, a word truncation approach that uses corpus statistics, and an elaborated lemmatizer-based stemmer provide similar performances in terms of IR effectiveness.

Most of the Turkish IR research studies developed their own stemmer. However, the most used stemmers in Turkish IR are Snowball (Porter, 2001) and Zemberek (Afsın and Dündar, 2007). Snowball is a stemmer developed using Snowball6 string processing language (Çilden, 2006) where Turkish words are analyzed with an affix stripping approach without any dictionary lookups. Zemberek is a stemmer designed exclusively for Turkish language. It is based on a root dictionary. Accordingly, it provides root forms of given words using a root dictionary-based parser combined with natural language processing algorithms. It handles special cases for suffixes and can be used as a lemmatizer based stemmer.

Yılmazel (2010) compared the retrieval performances of three retrieval models: Lemur TF.IDF, OKAPI and Language Modeling using the Milleyet test dataset. Two metrics were used to evaluate the retrieval performances: Mean Average Precision (MAP) and binary preference (bpref). As in Can et al. (2008), the author concluded that a stopword list does not have great effects on retrieval performance according to bpref values of different retrieval methods. The author stressed on the importance of the stemming and concluded that stemming can increase retrieval performances of the three models up to 20%. The best results are reached using the Zemberek stemmer. He also concluded that the retrieval performances of the three retrieval models are quite similar and no conclusion can be done regarding which model is more adequate for the Turkish language.

Earlier research has shown that using keyphrases to index Turkish documents with noun phrases improves Turkish IR. Pembe and Say (2004) based their work on the "Phrase Retrieval Hypothesis" (Arampatzis et al., 1998; 2000). They used a finite-state cascade technique to extract noun phrases from 615 Turkish documents collected from the Web. They used only 5 queries to evaluate their approach. Results showed retrieval improvement at low recall levels (10 and 20 documents). Arısoy et al. (2008) used feature sets that take the morphological characteristics of Turkish into account to extract bigrams and trigrams. Even if they applied the extracted N-grams to automatic speech recognition, their extraction approach have shown promising performances. Bahadir and Ilyas (2009) proposed a method for automating Turkish keyphrase extraction using multi-criterion ranking and compared their extracted keyphrases to manually assigned keyphrases but they did not apply their method within an IR system. Based on a statistical approach to extract words collocations, Metin and Karaoğlan (2010) concluded that Chi-square hypothesis test and mutual information methods have produced better results compared to other statistical methods on Turkish datasets. In a similar way, authors in Haddad and Bechikh Ali (2014) relied on Turkish morphological characteristics to extract compounds. They used compounds to index Turkish documents and queries. To extract compounds, the authors used only three syntactic patterns: < Noun Noun >, < Noun Adjective > and < Adjective Noun >. The best results are reached using the Zemberek stemmer. Using a more complete stopword list than the lists used in Aysın and Fazlı (1994) and in Can et al. (2008), the authors showed that removing stopwords can improve the retrieval performances. Their experiment results using compounds in addition to keywords to index queries and documents gave better IR performances than using only keywords.

In this article, we adopt a similar approach as in Haddad and Bechikh Ali (2014), but using a more complete compounds syntactic patterns list. Indeed, we use eight syntactic patters in our study; on the other hand, only three syntactic patterns were used in Haddad and Bechikh Ali (2014). We also combined different syntactic patterns in order to have a global view of compounds indexing for Turkish texts evaluation and performances. We also use the Zemberek stemmer and a stopword list as it is proved in the state-of-the-art that they improve the retrieval performances. In the next section, we present our methodology to extract compounds.

4. Our Turkish compounds extraction methodology

Our Turkish compounds extraction methodology uses a light linguistic approach based on three steps:

1. Using Zemberek, the stemming is applied on Milliyet dataset to produce stems. The stemming allows transforming inflected words forms in one same word (the stem). Indeed, the stemming processing of the three examples "*kapısının kolu*", "*kapının koludur*" and "*kapımızın kolunda*" will produce the same stemmed result "*kapı kol*".
2. Using Zemberek as an off-the shelf tagger, the system performs Part of Speech (POS) tagging to generate a tagged text from the original text. The POS is based on a set of lexical categories (e.g., nouns, adjectives, prepositions, gerundive, proper noun, determiner, etc.).
3. Using the tagged text, the compounds are extracted by the identification of syntactic patterns. We adopt the definition of syntactic patterns in Haddad (2003), where a pattern is a syntactic rule on the order of concatenation of grammatical categories which form a compound. Accordingly, we consider:

   - V: the vocabulary extracted from the corpus
   - C: a set of lexical categories
   - L: the lexicon $\subset V \times C$

   A pattern is then a syntactic rule of the form:

   $X := Y_1 Y_2 ... Y_n$ where $Y_i \in L$

In this study, we limited the syntactic patterns size to two components. For example, considering the sentence from the document number 400,000 of the Milliyet test dataset:

Kırmızı pasaportlara da kontenjan sınırı konuluyor

(The quota limit is being set on red passports)

First, the stemming process generates the following sentence:

kirmizi pasaport da kontejyan sinir konulmak

Then the POS tagging process generates the following tagged sentence:

Kırmızı/Adjective pasaport/Noun da/Preposition kontenjan/Noun sınır/Noun konulmak/Verb

Compounds are extracted by the identification of syntactic patterns presented in Section 2. From the above example, two compounds are extracted: the compound *Kırmızı pasaport* (Red passport) is extracted based on the syntactic pattern < Adjective Noun > and the compound *kontenjan sınırı* (Quota limit) is extracted based on the syntactic pattern < Noun Noun >. The extracted compounds are then used to index documents and queries.

5. Experimental setup

In this section, we describe the dataset, the IR system, the natural language processing framework, the stopword list and the evaluation metrics used in our study. We structure our findings by compound pattern types. For each pattern type, we give the evaluation performances then we give the results of the combined patterns. For example, the two patterns < Adjective Noun > and < Noun Noun > are evaluated individually, then the usage of the two patterns together is evaluated.

Our model is implemented upon the open-source Terrier IR platform version 4.0. The experimentation is composed of three main phases: (1) Compounds extraction, (2) Textual indexing and (3) IR results.

5.1. Milliyet dataset

Considering compounds indexing for Turkish texts evaluation, we have used Bilkent Information Retrieval Group Test dataset called Milliyet (Can et al., 2006; 2008). The dataset is about 800 Megabytes and 95.5 million words before stopwords elimination (Can et al., 2008). Without stopwords elimination, each document contains 234 words on average and 201 on average words after stopwords elimination. Figure 1 shows the document structure containing eight fields: document number, source, URL, date, time, author, headline (few words related to the document topic) and the document content (few sentences). The headline and document content contain textual information. We used only these two fields, in addition to the document number field to identify each document, in our experiments. We merged the two fields contents into one single field.

The Milliyet query structure was inspired from the TREC query structure. Indeed, in additional to the query number, the query structure contains three fields: topic (few words related to the query topic), description (few sentences) and narrative (explanation). In our experiments, we used and merged the two fields topic and description since using

< *DOC* >
< *DOCNO* > 15 < /*DOCNO* >
< *SOURCE* > *Milliyetv*.01 < /*SOURCE* >
< *URL* > *www.milliyet.com.tr*/2001/12/29/*ekonomi*/*eko*09.*html* < /*URL* >
< *DATE* > 2001/12/29/ < /*DATE* >
< *TIME* >< /*TIME* >
< *AUTHOR* >< /*AUTHOR* >
< *HEADLINE* >
Elektrik, *çay*, *şekervealoyazam*
< /*HEADLINE* >
< *TEXT* >
TEDAŞ, elektriğe aynıay içinde ikinci zammı yaptı. Aralık ayının baÅında yüzde 5.1 artırılan elektriğe, 24 Aralık'tan geçerli olmak üzere yüzde 2'lik yeni bir zam daha yapıldı. Böylece, elektriğe bir ay içinde yapılan toplam zam oranı yüzde 7'yi aÅştı.
< /*TEXT* >
< /*DOC* >

Fig. 1. Document number 15 of Milliyet dataset.

< *Query* >
< *QueryID* > 235 < /*QueryID* >
< *Topic* > Kuş Grib < /*Topic* >
< *Description* >
Kuş gribi nedir, nasıl bulaşır, belirtileri nelerdir sorularına cevap olabilecek dokümanlar.
< /*Description* >
< *Narrative* >
Kuş gribi ile alakalı her türlü bilginin elde edilebileceği bir doküman olmalı. Hastalığının tanımını, bulaşma yollarını, belirtilerini, varsa tedavi yollarını ve buna benzer birccok konuyu okuyucuya accıklayan nitelikte bir doküman.
< /*Narrative* >
< /*Query* >

Fig. 2. Query number 235 of Milliyet dataset.

the narrative field is deteriorating the retrieval performances (Haddad and Bechikh Ali, 2014). Figure 2 shows an example of the query structure.

The dataset have been built on an incomplete relevance judgments (pooling technique) (Spärck Jones and Van Rijsbergen, 1975). Indeed, to determine the relevant documents related to the queries, Bilkent Information Retrieval Group used the pooling concept. The queries are written and evaluated according to the TREC approach by 33 native speaker assessors. The original query owners performed the evaluation using binary judgments. During the evaluation, the query pool contents were presented to the assessors in random order, and the rest of the dataset was assumed to be irrelevant (Can et al., 2008).

5.2. Terrier IR system

The IR system used in both textual indexing and retrieval phases is Terrier (an open source IR framework). We use Terrier because it implements state-of-the-art indexing and retrieval functionalities and it has performed successfully in IR tasks in CLEF and TREC (Ounis et al., 2005). In our experiments, we use the three IR models: TF.IDF (Salton et al., 1975), BM25 (Robertson et al., 2004) and the Language Model (LM) (Ponte and Croft, 1998).

Milliyet dataset documents are short documents length (documents with maximum 100 words), medium documents length (documents with 101 to 300 words) and long documents length (documents with more than 300 words). As document length effects the retrieval performances, “ignore.low.idf” option is used in the Terrier system during our experiments in order not to ignore short documents.

5.3. Zemberek and stopword list

For Turkish text preprocessing, we use Zemberek library (an open source NLP framework for Turkic Language) (Afsın and Dündar, 2007). It’s officially used as spell checker in Open Office Turkish version and Turkish national Linux Distribution Pardus. Earlier works showed that it is performing better than other existing stemmers (Haddad and Bechikh Ali, 2014). We use Zemberek version 2.1.1 as a lemmatizer based stemmer and also as POS tagger. It uses a root dictionary-based parser.

Stopwords are applied using *Turkish StopWord List 1.1* provided by the Natural Language Processing Group, Department of Computer Engineering, Fatih University. The list contains 223 Turkish stopwords such as *acaba* (Just), *hepsi* (All) and *onu* (Him).

5.4. Retrieval evaluation metrics

Since the dataset is built on an incomplete relevance judgments, the Binary Preference metric (bpref) is the most adequate to evaluate retrieval performances. Indeed, bpref is based on the relative ranks of only the judged documents (Buckley and Voorhees, 2004).

To consider how retrieval bias relates to the metrics that consider both precision and recall, we include Mean Average Precision (MAP) to represent the overall effectiveness performance of our approach. We also include the 11-point interpolated average precision (11pt-avg). To evaluate our assumption that compounds increase Turkish IR precision, we use the precision at low recall considering only the top results returned by the system. Accordingly, we use Precision at cut-off level λ, for $\lambda = 5$ (P@5), 10 (P@10) and 15 (P@15).

6. Experimental results and discussion

In this section, we present and discuss the experimental results of the evaluation. To implement our model, we indexed unigram bag of words features in addition to compounds. The different experimentation results are presented in Table 3. The highest scores for each model/metric pair are shown using bold font.

6.1. Compound extraction results

To estimate the impact of different patterns, we first analyze how frequent the compounding phenomenon is in Milliyet test dataset. Table 2 provides details on compounds distribution. The most frequent compounds in the documents as well as in the queries are extracted using *Pattern1*, followed by *Pattern6* then *Pattern3*. For the pattern type $<$ Noun Adjective $>$, compounds extracted using *Pattern7* and *Pattern8* have a very low document frequencies (average respectively 0.37 and 0.23) as well as query frequencies (average respectively 0.04 and 0.03). However, this is not surprising owing to these patterns nature. Indeed, for *Pattern7*, the noun has to have third person singular possessive suffix. For *Pattern8*, the adjective is derived from a noun. Since the Zemberek tries to find the dictionary entry of a word, after applying the lemmatizer, these compounds are identified as *Pattern6* type.

A study was performed to evaluate our system compound extraction performance. A manual extraction was was performed on 20 documents from the Milliyet collection by a native Turkish speaker. 48 compounds were manually identified. The number of compounds extracted by the system is 37 compounds (70% of recall).

6.2. Baseline results

Since we are interested in measuring compounds indexing effectiveness, keyword indexing is used to have a baseline set of results. To determine whether our retrieval system was a reasonable choice as a baseline, we compared its results to two state-of-the-art performances that used the same Milliyet test dataset: Haddad and Bechikh Ali (2014) and Can et al. (2008). Haddad and Bechikh Ali (2014) used TF.IDF, BM25 and LM models. Can et al. (2008) used only the TF.IDF model. As can be seen from Table 3, using the preprocessing (stopwords list, Zemberek, etc.) results in better performances compared to the state-of-the-art results. Indeed, precision of the baseline cut-off points P@5 and P@10 performances are higher than Haddad and Bechikh Ali (2014) and Can et al. (2008) performances. We can observe the same for the 11pt-avg and the bpref metrics. For the MAP metric, Can et al. (2008) approach performed slightly better then our baseline. We also compare our results to bigram indexing (Lioma, 2008), this run is called bigram.

We performed statistical significance tests to establish whether the differences between means of evaluation metrics are significant or should be attributed to chance and significant differences were found with $p < 0.05$ among the compounds use. MAP, bpref and 11pt-avg results of the baseline and the bigram results are compared using the bilateral paired Student t-test (Smucker et al., 2007).

Table 2
Compounds extraction statistics.

Pattern	# of compounds in Documents	Average Compounds per Document	# of compounds in Queries	Average Compounds per Query
Pattern1	2,374,217	5.18	143	1.99
Pattern2	172,657	0.42	12	0.17
Pattern3	1,013,562	2.48	26	0.36
Pattern4	220,545	0.54	14	0.19
Pattern5	196,759	0.48	14	0.19
Pattern6	1141,923	2.80	35	0.49
Pattern7	151,165	0.37	3	0.04
Pattern8	95,341	0.23	2	0.03

Table 3
Retrieval performances evaluation based on syntactic patterns. The symbols ↑ and * denote significant MAP, Bref and 11pt-avg difference based on, respectively, the baseline run and Bigram run (t-test, p ≤ 0:05).

TF.IDF Run	P@5	P@10	P@15	11pt-avg	MAP	bpref
Baseline	0.6500	0.6097	0.5889	0.3734	0.3556	0.4739
Haddad and Bechikh Ali (2014)	0.5694	0.5556	0.5259	0.3518	0.3351	0.4407
Can et al. (2008)	–	0.5917	–	–	**0.4092**	0.4322
bigram	0.6583	**0.6194**	**0.5981**	0.3758	0.3592	0.4713
Pattern1	0.6583	0.6069	0.5889	0.3738	0.3560	0.4738
Pattern2	0.6500	0.6097	0.5880	0.3735	0.3556	0.4738
Pattern3	0.6000	0.5847	0.5620	0.3509	0.3322	0.4523
Pattern4	0.6528	0.6097	0.5889	0.3734	0.3556	0.4739
Pattern5	0.6500	0.6111	0.5889	0.3724	0.3554	0.4739
Pattern6	0.6111	0.5903	0.5648	0.3485	0.3315	0.4525
Pattern7	0.6167	0.5889	0.5676	0.3504	0.3334	0.4535
Pattern8	0.6083	0.5903	0.5676	0.3506	0.3336	0.4534
Noun-Noun	0.6639	0.6069	0.5889	0.3738	0.3559	0.4737
Adject-Noun	0.6556	0.6097	0.5898	0.3725	0.3556	0.4735
All patterns	0.6583	0.6097	0.5907	0.3734	0.3563	0.4733
Patterns 1, 4 and 7	**0.6667**	0.6139	0.5917	**0.3768**$^{\uparrow}$	0.3593$^{\uparrow}$	**0.4745**

BM25 Run	P@5	P@10	P@15	11pt-avg	MAP	bpref
Baseline	0.6111	0.5931	**0.5898**	**0.3864**	0.3691	0.4625
Haddad and Bechikh Ali (2014)	0.5694	0.5486	0.5306	0.3129	0.2934	0.4472
bigram	0.6083	0.5972	0.5870	0.3862	0.3680	0.4793
Pattern1	0.6472	0.6111	0.5833	0.3826	0.3663	0.4810$^{\uparrow *}$
Pattern2	0.6417	0.6097	0.5833	0.3820	0.3659	0.4809$^{\uparrow *}$
Pattern3	0.6028	0.5889	0.5667	0.3566	0.3418	0.4594
Pattern4	**0.6528**	0.6083	0.5815	0.3821	0.3659	0.4808$^{\uparrow}$
Pattern5	0.6472	0.6083	0.5824	0.3821	0.3659	0.4809$^{\uparrow *}$
Pattern6	0.6167	0.5958	0.5676	0.3559	0.3416	0.4594
Pattern7	0.6222	0.6014	0.5657	0.3587	0.3438	0.4604
Pattern8	0.6167	0.5944	0.5676	0.3585	0.3440	0.4603
Noun-Noun	0.6444	0.6111	0.5824	0.3829	0.3662	0.4809$^{\uparrow *}$
Adject-Noun	0.6444	0.6097	0.5787	0.3820	0.3663	0.4809$^{\uparrow *}$
All patterns	0.6444	0.6125	0.5824	0.3827	0.3667	0.4625*
Patterns 1, 4 and 7	0.6444	**0.6153**	0.5824	0.3855	**0.3692**	**0.4819**$^{\uparrow *}$

LM Run	P@5	P@10	P@15	11pt-avg	MAP	bpref
Baseline	0.5639	0.5653	0.5472	**0.3827**	**0.3681**	0.4333
Haddad and Bechikh Ali (2014)	0.5694	0.5167	0.5065	0.3227	0.3030	0.4086
bigram	0.5667	0.5639	0.5407	0.3735	0.3569	0.4291
Pattern1	0.5806	0.5736	**0.5537**	0.3791	0.3627*	0.4358*
Pattern2	0.5861	0.5806	0.5472	0.3783	0.3623*	0.4356*
Pattern3	0.5417	0.5208	0.5046	0.3494	0.3339	0.4110
Pattern4	0.5889	**0.5819**	0.5519	0.3787	0.3626*	0.4357*
Pattern5	**0.5917**	0.5806	0.5481	0.3785	0.3625*	0.4358*
Pattern6	0.5389	0.5250	0.5111	0.3507	0.3339	0.4108
Pattern7	0.5361	0.5306	0.5139	0.3512	0.3352	0.4116
Pattern8	0.5361	0.5278	0.5093	0.3513	0.3350	0.4114
Noun-Noun	0.5833	0.5736	0.5509	0.3793	0.3627*	0.4359*
Adject-Noun	0.5861	0.5778	0.5481	0.3787	0.3625*	0.4356
All patterns	0.5611	0.5639	0.5435	0.3822	0.3676*	0.4325
Patterns 1, 4 and 7	0.5861	0.5764	0.5519	0.3764	0.3659*	**0.4379***

6.3. < Noun Noun > patterns

For the TF.IDF, *Pattern1* and *Pattern2* achieved comparable performances compared to the baseline run and the bigram run. Combining *Pattern1* and *Pattern2* performed slightly better than the bigram run for the MAP and the bpref measures.

For the BM25, *Pattern1* and *Pattern2* performed better than the baseline run and the bigram run for P@5 and P@10 measures. *Pattern1* increased the P@5 measure by 5.9% compared to the baseline run and by 6.4% compared to the bigram run. Combining *Pattern1* and *Pattern2* achieved better performances compared to the baseline run and the bigram run for the P@5 and P@10 measures. *Pattern1, Pattern2* and the combination of *Pattern1, Pattern2* significantly improved the bpref measure compared to the baseline run and the bigram run bpref performances.

For the LM at P@5, *Pattern1* and *Pattern2* performed better than the baseline and the enhancement was respectively about 2.9% and 3.9%. *Pattern1* and *Pattern2* achieved a P@5 increase of 2.4% and 3.4% compared to the bigram run. Both *Pattern1* and *Pattern2* significantly improved MAP and bpref measures compared to the bigram run.

Combining *Pattern1* and *Pattern2* results in a better performances compared to the baseline run and the bigram run results for the precisions at low recall. Nevertheless, the MAP and bref performances of the combination are the same as *Pattern2* performances.

6.4. < Adjective Noun > patterns

Applying the TF.IDF model, *Pattern4* and *Pattern5* performed comparably to the baseline but *Pattern3* performed less than the baseline. Even if the extracted compound number using *Pattern3* is more important than extracted compound numbers using *Pattern4* and *Pattern5*, indexing documents and queries with compounds extracted using *Pattern3* is deteriorating the results compared to the baseline.

For the BM25 model, *Pattern4* and *Pattern5* performed better than the baseline run and the bigram run on P@5 and P@10 measures. *Pattern4* increased the P@5 measure by 6.8% compared to the baseline run and by 7.3% compared to the bigram run. *Pattern4* and *Pattern5* outperformed the baseline run to a statistically significant degree for the bpref.

For the LM model, *Pattern4* and *Pattern5* performed better than the baseline run and the bigram run on precision measures and significantly improved MAP and bpref compared to the bigram baseline.

Combining *Pattern3, Pattern4* and *Pattern5* achieved comparable results to the baseline run and bigram run for the TF.IDF model and BM25 model.

6.5. < Noun Adjective > patterns

Results in Table 3 show that indexing with < Noun Adjective > patterns is less effective than indexing with < Noun Noun > patterns and < Adjective Noun > patterns. This due to the fact that few compounds from this type exist in the queries as shown in Table 1. *Pattern6, Pattern7* and *Pattern8* results are quite similar with a slight advantage for *Pattern7*.

6.6. Results discussion

Compared to the baseline performances on P@5, P@10 and P@15 precision measures, better results have been achieved by *Pattern1, Pattern2, Pattern4* and *Pattern5. Pattern1* and *Pattern5* achieved the best performances.

The experiment version combining *Pattern1, Pattern4* and *Pattern7* is the best overall version as shown in Table 3. TF.IDF model achieved results that significantly improved MAP and 11pt-avg compared to the baseline run. Further, it achieved the best P@5 measure performance.

BM25 model provided the best global metrics MAP and bpref performances, where statistically significant improvement was achieved compared to the baseline run and the bigram run on the bpref measure.

Combining < Noun,Noun > patterns performed comparably to the < Noun Adjective > patterns. Compared to the baselines performances, using compounds indexing increased the precision at low recall for the BM25 model when combining all the patterns.

The experiment version combining Pattern1, Pattern4 and Pattern7 outperformed the baseline performances at all document cut-off values examined for the three models.

7. Limitations and future work

This article presented an empirical evaluation of compounds indexing for Turkish texts. Due to the patterns ambiguity, we exclude *Patterns* 9, 10, 11 and 12 from our experiments. Indeed, a deep linguistics processing is needed to extract these compound types.

Compared to the baseline and the state-of-the art performances, experiments on Turkish Milliyet IR dataset using compounds to index documents and queries showed significant improvement. Our experiments verify our expectation: improving Turkish retrieval system performances by using compounds indexing, especially precision performances at low recall levels.

The work reported here is still preliminary and further experiments are required to understand possible compounds effects between the combined improvements. This is a challenge that remains open and we expect that our approach performs better if applied to a technical dataset like the medical domain for example. Indeed, compounds in technical domains are an important indexing resource. Furthermore, our experiments are based on compounds extracted using syntactic patterns but other compound have free constituent order in syntax (Göknel, 2010).

For the experiments described above we used a weighting schemata which do not differentiate between words and compounds. We intend to investigate the effect of an alternative weighting schema in the future. Further experiments on Turkish sentiments analysis datasets will be carried out.

References

Afsın, A., Dündar, A., 2007. Coreferentiality and stress. Structure 10, 1–5.

Alpkocak, A., Ceylan, M., 2012. Effects of diacritics on Turkish information retrieval. Turk. J. Electr. Eng. Comput. Sci., The ScI. Technol. Res. Counc. Turk. 20 (5), 787–804.

Arampatzis, A., Tsoris, T., Koster, C.H.A., van der Weide, T.P., 1998. Phrase-based information retrieval. Inf. Process. Manag. 34 (6), 693–707.

Arampatzis, A., Van der Weide, T.P., Koster, C., Van Bommel, P., 2000. An evaluation of linguistically-motivated indexing schemes. In: Proceedings of the 22nd BCS-IRSG Colloquium on IR Research. Cambridge, England.

Arısoy, E., Roark, B., Shafran, I., Saraçlar, M., 2008. Discriminative n-gram language modeling for Turkish. In: Proceedings of the Speech Communication Association. Brisbane, Australia, pp. 825–828.

Arslan, A., 2016. Deasciification approach to handle diacritics in turkish information retrieval. Inf. Process. Manage. 52 (2), 326–339.

Aysın, S., Fazlı, C., 1994. Effects of stemming on Turkish text retrieval. In: Proceedings of the Ninth International Symposium on Computer and Information Sciences. Antalya, Turkey, pp. 49–56.

Bahadir, O., Ilyas, C., 2009. Turkish keyphrase extraction using multi-criterion ranking. In: Proceedings of the Computer and Information Sciences. Guzelyurt, Cyprus, pp. 269–273.

Bechikh Ali, C., Rui, W., Hatem, H., 2015. A two-level keyphrase extraction approach. In: Proceedings of the Computational Linguistics and Intelligent Text Processing. Cairo, Egypt, pp. 390–401.

Bendersky, M., Croft, W.B., 2008. Discovering key concepts in verbose queries. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2008, Singapore, July 20-24, 2008, pp. 491–498.

Bendersky, M., Metzler, D., Croft, W.B., 2011. Parameterized concept weighting in verbose queries. In: Proceeding of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2011, Beijing, China, July 25-29, 2011, pp. 605–614.

Buckley, C., Singhal, A., Mitra, M., 1995. New retrieval approaches using smart: Trec 4. In: Proceedings of the Fourth Text REtrieval Conference, Maryland, USA.

Buckley, C., Voorhees, E.M., 2004. Retrieval evaluation with incomplete information. In: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, United Kingdom, pp. 25–32.

Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H.C., Vursavas, O.M., 2006. First large-scale information retrieval experiments on Turkish texts. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Washington, USA, pp. 627–628.

Can, F., Kocberber, S., Balcik, E., Kaynak, C., Ocalan, H.C., Vursavas, O.M., 2008. Information retrieval on Turkish texts. J. Assoc. Inf. Sci. Technol. 59 (3), 407–421.

Çilden, E.K., 2006. Snowball: stemming Turkish words using snowball. Http://snowball.tartarus.org.

Ekmekçioglu, F.C., Willett, P., 2000. Effectiveness of stemming for turkish text retrieval. PROGRAM 34 (2), 195–200.

Fagan, J.L., 1987. Automatic phrase indexing for document retrieval: an examination of syntactic and non-syntactic methods. In: Proceedings of the Tenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Louisiana, USA, pp. 91–101.

Göknel, Y., 2010. Turkish Grammar (Transformational Generative and Contrastive). Vesta Yayinlari, Istanbul.

Göksel, A., Haznedar, B., 2007. Remarks on compounding in Turkish. In: Part of the MorboComp Project, University of Bologna.

Gupta, M., Bendersky, M., 2015. Information retrieval with verbose queries. Found. Trends Inf. Retr. 9 (3-4), 91–208.

Haddad, H., 2003. French noun phrase indexing and mining for an information retrieval system. In: Proceedings of the String Processing and Information Retrieval. Manaus, Brazil, pp. 277–286.

Haddad, H., Bechikh Ali, C., 2014. Performance of Turkish information retrieval: evaluating the impact of linguistic parameters and compound nouns. In: Proceedings of the Computational Linguistics and Intelligent Text Processing. Kathmandu, Nepal, pp. 381–391.

Hafer, M.A., Weiss, S.F., 1974. Word segmentation by letter successor varieties. Inf. Storage Retr. 10 (11-12), 371–385.

Köksal, A., 1981. Tümüyle Özdevimli deneysel bir belge dizinleme ve erisim dizgesi: TÜder. In: Proceedings of the Ulusal Bilisim Kurultayi. Ankara, Turkey, pp. 37–44.

Kornfilt, J., 1997. Turkish (Descriptive Grammars). Kindle Edition, New York, USA.

Lioma, C., 2008. Part of speech N-grams for information retrieval. Ph.D. thesis. University of Glasgow, UK.

Lioma, C., Simonsen, J.G., Larsen, B., Hansen, N.D., 2015. Non-compositional term dependence for information retrieval. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 595–604.

Metin, S.K., Karaoğlan, B., 2010. Collocation extraction in Turkish texts using statistical methods. In: Proceedings of the Advances in Natural Language Processing, Reykjavik, Iceland, pp. 238–249.

Metzler, D., Croft, W.B., 2005. A Markov random field model for term dependencies. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 472–479.

Mitra, M., Buckley, C., Singhal, A., Cardie, C., 1997. An analysis of statistical and syntactic phrases. In: Proceedings of the Computer-Assisted Information Retrieval. Montreal, Canada, pp. 200–217.

Oflazer, K., 2014. Turkish and its challenges for language processing. Lang. Resour. Eval. 48 (4), 639–653.

Ounis, I., Amati, G., Plachouras, V., He, B., CMacdonald, r., Johnson, D., 2005. Terrier information retrieval platform. In: Proceedings of the European Conference on IR Research, Santiago de Compostela, Spain, pp. 517–519.

Park, J.H., Croft, W.B., Smith, D.A., 2011. A quasi-synchronous dependence model for information retrieval. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 17–26.

Pembe, F.C., Say, A.C.C., 2004. A linguistically motivated information retrieval system for Turkish. In: Proceedings of the Computer and Information Sciences, Antalya, Turkey, pp. 741–750.

Pickens, J., Croft, W.B., 2000. An exploratory analysis of phrases in text retrieval. In: Proceedings of the Computer-Assisted Information Retrieval, College de France, France, pp. 1179–1195.

Ponte, J.M., Croft, W.B., 1998. A language modeling approach to information retrieval. In: Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Melbourne, Australia, pp. 275–281.

Porter, M.F., 2001. Snowball: a language for stemming algorithms. Http://snowball.tartarus.org.

Preslav, N., 2013. On the interpretation of noun compounds: syntax, semantics, and entailment. Nat. Lang. Eng. 19 (3), 291–330.

Robertson, S.E., Zaragoza, H., Taylor, M.J., 2004. Simple BM25 extension to multiple weighted fields. In: Proceedings of the Conference on Information and Knowledge Management, Washington, DC, USA, pp. 42–49.

Salton, G., Yang, C.S., Yu, C.T., 1975. A theory of term importance in automatic text analysis. J. Am. Soc. Inf. Sci. 26 (1), 33–44.

Sanderson, M., 1994. Word sense disambiguation and information retrieval. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, pp. 142–151.

Sever, H., Bitirim, Y., 2003. Findstem: analysis and evaluation of a turkish stemming algorithm. In: Proceedings of the String Processing and Information Retrieval. Manaus, Brazil, pp. 238–251.

Smucker, M.D., Allan, J., Carterette, B., 2007. A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of the Sixteenth ACM Conference on Information and Knowledge Management, CIKM 2007, Lisbon, Portugal, November 6-10, 2007, pp. 623–632.

Solak, A., Can, F., 1994. Effects of stemming on turkish text retrieval. In: Proceedings of the Computer and Information Sciences, Antalya, Turkey, pp. 49–56.

Spärck Jones, K., Van Rijsbergen, C., 1975. Report on the need for and provision of an "Ideal" information retrieval test collection. Technical Report, 5266. British Library Research and Development, University of Cambridge.

Strzalkowski, T., Lin, F., Wang, J., Perez-Carballo, J., 1999. Evaluating natural language processing techniques in information retrieval. In: Proceedings of the Natural Language Information Retrieval, pp. 113–145.

Tong, X., Zhai, C., Milic-Frayling, N., Evans, D.A., 1996. Evaluation of syntactic phrase indexing − CLARIT NLP track report. In: Proceedings of The Fifth Text REtrieval Conference. Maryland, USA, pp. 347–358.

Trask, R., 1993. A Dictionary of Grammatical Terms in Linguistics. Routledge, New York, USA.

Yılmazel, O., 2010. A language modeling approach to turkish text retrieval. J. Sci. Technol. A- Appl. Sci. Eng., Anadolu Univ. 11 (2), 163–172.

Zhang, W., Liu, S., Yu, C., Sun, C., Liu, F., Meng, W., 2007. Recognition and classification of noun phrases in queries for effective retrieval. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 711–720.