

Word Sketches for Turkish

Bharat Ram Ambati, Siva Reddy, Adam Kilgarriff

Lexical Computing Ltd, UK

bharat.ambati@gmail.com, siva@sketchengine.co.uk, adam@lexmasterclass.com

Abstract

Word sketches are one-page, automatic, corpus-based summaries of a word’s grammatical and collocational behaviour. In this paper we present word sketches for Turkish. Until now, word sketches have been generated using a purpose-built finite-state grammars. Here, we use an existing dependency parser. We describe the process of collecting a 42 million word corpus, parsing it, and generating word sketches from it. We evaluate the word sketches in comparison with word sketches from a language independent sketch grammar on an external evaluation task called topic coherence, using Turkish WordNet to derive an evaluation set of coherent topics.

Keywords: Word Sketches, Turkish, Sketch Grammar, Dependency Parsing, Topic Coherence

1. Introduction

Word sketches are one-page, automatic, corpus-based summaries of a word’s grammatical and collocational behaviour. They were first used in the production of the Macmillan English Dictionary (Rundell, 2002). At that point, word sketches only existed for English. Today, they are built into the Sketch Engine (Kilgarriff et al., 2004), a corpus tool which takes as input a corpus of any language and generates word sketches for the words of that language. It also automatically generates a thesaurus and ‘sketch differences’, which specify similarities and differences between near-synonyms.

Turkish is the 21st largest language in the world, with over 50m speakers¹, yet until recently there were few language resources available for it (Oflazer, 1994). The last decade has seen much increased activity with new tools such as a morphological analyzer and disambiguator (Yuret and Ture, 2006) and dependency parser (Eryiğit et al., 2008).

We first gathered the corpus from the web using the ‘Corpus Factory’ as described in (Kilgarriff et al., 2010b), then cleaned and deduplicated it using the jusText and Onion tools (Pomikálek, 2011), then lemmatized and POS-tagged it with Yuret and Ture’s tool. Up until now, the next step would have been to load it into the Sketch Engine, and to prepare a ‘sketch grammar’ which would be used for finite-state shallow parsing to identify grammatical relations. However for Turkish we did not have an expert available to write that grammar: what was available was a parser (which we would also expect to be more accurate). So, instead, we extended the Sketch Engine input formalism so that it could accept parser output in CONLL format². Then we generate word sketches directly from the parser output. Here we present these first word sketches for Turkish, which are also the first word sketches to be the product of a parser.

2. TurkishWaC: A Turkish web corpus of 42 million words

The corpus was collected using the Corpus Factory method (Kilgarriff et al., 2010b). First, we gather a list of ‘seed words’ of the language from its Wikipedia³. Then we generate several thousand search engine queries by randomly selecting three seed words. We then send these queries to a commercial search engine (in this case, Bing⁴). We then gather all the pages that Bing identifies in its hits pages. The pages are filtered using a language model, and body text extraction, deduplication and encoding normalization are performed thus building a clean corpus. We replaced body-text extraction and deduplication tools with the state-of-art tools jusText and Onion respectively (Pomikálek, 2011). The final corpus, TurkishWaC⁵, is of size 42.2 million word and is accessible within the Sketch Engine⁶.

3. TurkishWaC Annotation

In this section, we first describe some relevant linguistic properties of Turkish, and then we describe different tools used to process TurkishWaC.

Turkish is an agglutinative language with rich morphology. Turkish words may be formed through very productive processes, and may have many inflected forms. The morphological structure of a Turkish word may be represented by splitting the word into inflectional groups (IGs). The root and derivational elements of a word are represented by different IGs, separated from each other by derivational boundaries (DB). Each IG will have its own part of speech and inflectional features. An example taken from (Eryiğit et al., 2008) is shown below.

	arabanızdayı	
	(‘it was in your car’)	
araba+Noun+A3sg+P2pl+Loc	arabanızda	DB ydı
	DB	+Verb+Zero+Past+A3sg
$\underbrace{\hspace{1cm}}$	$\underbrace{\hspace{1cm}}$	$\underbrace{\hspace{1cm}}$
IG_1	‘in your car’	IG_2
		‘it was’

³<http://dumps.wikimedia.org/trwiki>

⁴<http://bing.com>

⁵WaC stands for the acronym Web as Corpus.

⁶<http://sketchengine.co.uk>

¹<http://www.ethnologue.com> (accessed October 2011)

²<http://ilk.uvt.nl/conll/>

ID	WORD	LEMMA	POSTAG	HEAD	DEPREL
1	Eğer	eğer	Conj	13	S.MODIFIER
2	ki	ki	Conj	1	INTENSIFIER
3	ülkelere	ülke	Noun	4	OBJECT
4	ve	ve	Conj	12	COORDINATION
5	onların	o-p	Pron	8	SUBJECT
6	özelliklerine	özellik	Noun	8	DATIVE.ADJUNCT
7	ilginiz	ilgi	Noun	8	SUBJECT
8	varsıa	var	Verb	12	MODIFIER
9	bu	bu	Det	10	DETERMINER
10	bölüm	bölüm	Noun	12	SUBJECT
11	ilginizi	ilgi	Noun	12	OBJECT
12	çekebilir	çek	Verb	13	SENTENCE
13	.	.	Punc	0	ROOT

Figure 1: A sample output of the parser in CONLL format

Turkish is a flexible constituent order language. Though the predominant order is SOV, constituents can freely change their position according to the requirements of the discourse context. It has been suggested that free-word order languages can be handled better using a dependency framework rather than a constituency-based one (Hudson, 1984; Shieber, 1985).

We needed a morphological analyzer which accounted for this rich morphology. Oflazer (1994) describes such an analyzer. It is a two-level analyzer which produces derivational boundary (DB) and inflectional groups (IGs). It gives different possible morphological analyses, including part-of-speech (POS) tags, for each word. We first converted from UTF-8 (the encoding in which TurkishWaC had been prepared) into latin-5 (as required for the tools we were to use). We then applied Oflazer’s morphological analyzer to the corpus. Out of the multiple analyses that were output, we needed to select the contextually correct one for each word. We used the morphological disambiguator of Yuret and Ture (2006) which has an accuracy of 96% for this purpose. For a word not recognized by the morphological analyzer, we first checked if it was either a punctuation mark or a number and, if it was, assigned the corresponding POS tag. For the rest, we tagged them as proper nouns.

Eryiğit et al. (2008) used MaltParser (Nivre and Hall, 2005) trained on a Turkish dependency treebank data for parsing Turkish. MaltParser is a system for data-driven dependency parsing, which can be used to induce a parsing model from treebank data and to parse new data using an induced model. We selected Nivre Arc-Standard algorithm of MaltParser as it gave the best accuracy for Turkish language. Eryiğit et al. (2008) showed that using IGs as the basic parsing units rather than words improved parser performance. So, we used IGs as basic parsing units.

Figure 1 displays a sample output of Turkish parser in CONLL format. On a quadcore system, it took 10 days to parse the whole TurkishWaC.

4. Word Sketches from TurkishWaC

The first step in generating word sketches is to generate dependency tuples. To date, Sketch Engine generates these

Sentence

We/PRP created/VB the/DET first/ADJ word/NN sketches/NN for/PREP Turkish/NN

Sketch Grammar

OBJECT:

1:[tag="VB"] [tag="DET"]{0,1} [tag="ADJ"] × [tag="NN"] 2:[tag="NN"]

Figure 2: Sketch Grammar for OBJECT relation

tuples from a corpus using Sketch Grammar. For example, take the sentence and the sketch grammar displayed in Figure 2. The grammar rule means that the word with tag VB is in relation OBJECT with the word with tag NN, if VB is followed by an optional DET tag followed by any number of ADJs and NNs. This grammar rule generates the dependency tuple (*sketches*, *OBJECT*, *created*), which means that *sketches* is the *OBJECT* of *created*.

(ki, INTENSIFIER, eğer), (ülkelere, OBJECT, ve),
(ve, COORDINATION, çek), (o, SUBJECT, var),
(özellik, DATIVE.ADJUNCT, var),
(ilgi, SUBJECT, var), (var, MODIFIER, çek),
(bu, DETERMINER, bölüm), (bölüm, SUBJECT, çek),
(ilgi, OBJECT, çek)

Figure 3: Dependency tuples from Figure 1

4.1. Word Sketches using Turkish dependency parser

Since Turkish had an existing parser which provides dependency information, we aim to make use of parser’s output rather than writing a sketch grammar to generate dependency tuples. In figure 1, the column HEAD denotes that the current word is in relation DEPREL with the word whose column ID is equal to HEAD. For example, the lemma *ilgi* (ID:7) is the SUBJECT (column DEPREL) of the lemma *var* (ID:8). All the tuples generated from the sentence in Figure 1 are displayed in Figure 3. Apart from

MODIFIER	861	1.1	OBJECT_OF	1143	2.9	SUBJECT_OF	463	1.6
kepek	<u>19</u>	9.18	piş	<u>31</u>	7.28	doğra	<u>6</u>	6.76
maya	<u>25</u>	8.58	ban	<u>9</u>	6.86	piş	<u>19</u>	6.73
bayat	<u>10</u>	8.41	kızar	<u>9</u>	6.84	lezzet	<u>4</u>	5.38
dilim	<u>33</u>	8.02	doğra	<u>5</u>	6.04	kon	<u>6</u>	2.93
pastırma	<u>4</u>	7.11	muhtaç	<u>5</u>	5.66	ye	<u>19</u>	2.79
nohut	<u>6</u>	7.07	ye	<u>104</u>	5.23	sat	<u>6</u>	2.61
Et	<u>5</u>	6.9	dene	<u>11</u>	4.87	kız	<u>5</u>	2.31
beyaz	<u>24</u>	6.7	götür	<u>17</u>	4.39	üre	<u>5</u>	2.08
tok	<u>4</u>	6.56	böl	<u>8</u>	4.17	yok	<u>6</u>	0.78
taze	<u>7</u>	6.56	dol	<u>9</u>	3.97	ne	<u>5</u>	0.46

Figure 4: Word Sketch of *ekmek* (*bread*) from dependency parser

noun_left	4538	0.9	adj_left	1079	1.1	verb_right	1269	1.5
fırın	<u>56</u>	6.99	kepek	<u>24</u>	8.13	piş	<u>27</u>	5.41
ekmek	<u>90</u>	6.65	bayat	<u>11</u>	6.99	doğra	<u>5</u>	4.75
dilim	<u>55</u>	6.54	pastırma	<u>6</u>	6.82	lezzet	<u>5</u>	3.47
yufka	<u>16</u>	6.46	maya	<u>26</u>	6.18	sevin	<u>5</u>	3.09
nohut	<u>24</u>	6.38	tok	<u>6</u>	5.29	dene	<u>8</u>	2.77
kepek	<u>15</u>	6.29	piş	<u>24</u>	5.25	ye	<u>81</u>	2.59
buğday	<u>49</u>	6.27	nohut	<u>6</u>	5.04	dağıt	<u>5</u>	2.23
som	<u>12</u>	5.97	taze	<u>13</u>	4.98	götür	<u>9</u>	2.11
lavaş	<u>9</u>	5.95	beyaz	<u>32</u>	4.18	böl	<u>6</u>	1.98

Figure 5: Word Sketch of *ekmek* (*bread*) from universal sketch grammar

these, we also generate additional tuples depending upon the type of relation like symmetric (e.g. COORDINATION), dual (e.g. OBJECT/OBJECT_OF), unary (e.g. INTRANSITIVE), trinary (e.g. PP_IN).

Once these tuples are generated, we rank all its collocations (words in relation with the target word) in each grammatical relation using logDice (Curran, 2004; Rychlý, 2008) and create a word sketch for a target word.

The word sketches of the word *ekmek* (*bread*) for selected grammatical relations are displayed in Figure 4.

4.2. Universal Sketch Grammar

Recently, we designed a sketch grammar which can be applied for any corpora irrespective of the language, and so is the name Universal Sketch Grammar. The grammar aims to capture word associations of a given word. We define relation names based on the location of the context words w.r.t. the target word. For example, all the verbs located left to a word within a distance of three from the target word are in the relation verb_left with the target word. The grammar describing this rule is

```
=verb_left
2:[tag="V.*"] [tag=".*"]{0,3} 1:[]
```

Similarly we define the relations verb_right, noun_left, noun_right, adjective_left, adjective_right, adverb_left and

adverb_right. Additionally we define the relations nextleft and nextright for the words immediately next to a given word. We also capture conjunction using the following rule.

=conj

```
1:[] [tag="C.*"] 2:[]
```

Figure 5 display the word sketches from universal sketch grammar.

5. Thesaurus from Word Sketches

In Sketch Engine, distributional thesaurus can be built for any language if the word sketches of the language exist. The thesaurus is built by computing similarity between words based on the extent of overlap between their word sketches. In contrast to earlier approaches of building a distributional thesaurus (Lin, 1998), Sketch Engine's implementation (Rychlý and Kilgarriff, 2007) is known for its speed with most thesauri computation taking less than an hour. The thesaurus can also cluster similar words into different groups which share common meaning. Since word sketches for Turkish exist, we have also built its distributional thesaurus. Figures 6 and 7 display the distributional thesaurus entries of the word *ekmek* (*bread*) from dependency parser and universal sketch grammar.

Lemma	Score	Freq
yemek	0.198	8923
yiyecek	0.17	1559
meyve	0.157	4279
süt	0.15	5093
yumurta	0.149	3052
kahve	0.139	2027
şarap	0.133	1472
sebze	0.132	1693
balık	0.131	5524
hamur	0.129	1115
peynir	0.128	912
yoğurt	0.123	1253

Figure 6: Thesaurus entry of *ekmek* (*bread*) from dependency based word sketches

Lemma	Score	Freq
yemek	0.289	8923
piş	0.28	2268
sebze	0.235	1693
peynir	0.232	912
tatlı	0.226	4045
meyve	0.225	4279
hamur	0.222	1115
şeker	0.213	3341
tatlı	0.209	2352
buğday	0.208	1804
yoğurt	0.206	1253
yumurta	0.203	3052

Figure 7: Thesaurus entry of *ekmek* (*bread*) from universal sketch grammar

6. Evaluation

The typical evaluation of word sketches is performed manually by lexicographers who are native speakers of the target language. A sample of words is chosen for evaluation, and word sketches for these words are evaluated by lexicographers who assess, for each collocation, whether they would include it in a published collocations dictionary (Kilgarriff et al., 2010a). The higher the average score over all the collocations, the higher is the accuracy of the word sketches. However in the case of Turkish, we did not have access to lexicographers.

Instead, we opted for an automatic evaluation of word sketches. Reddy et al. (2011) used word sketches in an external task called semantic composition. Inspired from it, we evaluate word sketches on an another external task, the task of topic coherence (Newman et al., 2010). A topic is a bag of words which are similar to each other and describe a coherent theme. In the task of topic coherence, given a topic, we score the topic for its coherence. The higher the similarity between words in the topic, the higher is the coherence. To find the similarity between two words, we make use of thesauri generated from word sketches. Our intuition is that for a given coherent topic, the topic coherence score predicted by a thesaurus generated from high quality word sketches is higher than the score from a thesaurus generated from low quality word sketches.

Distance	Noun	Verb	Adjective
Thesaurus from dependency parser sketches			
0	0.007843	0.012402	0.001504
1	0.005597	0.011392	0.005637
2	0.004768	0.014402	0.004523
Thesaurus from Universal Sketch Grammar			
0	0.006562	0.009519	0.007224
1	0.005672	0.008972	0.007784
2	0.004532	0.011920	0.006844

Table 1: Topic coherence scores of thesauri over WordNet

6.1. Coherent Topic Selection

We use Turkish WordNet to choose coherent topics. A wordnet synset (a synonym set) represents a highly coherent topic since all the words in the synset describe an identical meaning (topic). In WordNet, synsets are arranged in hierarchy in which a synset is linked with its hypernyms, hyponyms, antonyms, meronyms, holonyms etc. A synset along with its linked synsets at a distance of one or two also represent a topic, but with a different degree of coherence.

A topic built from a synset S and its related synsets at a distance d can be formally represented as a set of words $T = \{w_i : w_i \in S^*\}$, where S^* represents the union of the synset S and its related synsets. $S^* = \bigcup S_i$ for all S_i s.t. $\text{distance}(S, S_i) \leq d$

6.2. Topic Coherence Score

For a given topic $T = \{w_1, w_2, \dots, w_n\}$, we calculate its coherence by taking the average similarity over all the pairs of words in T .

$$C_T = \frac{\sum_{i,j} \text{sim}(w_i, w_j)}{n * (n - 1)/2}$$

where $\text{sim}(w_i, w_j)$ represents the thesaurus similarity between the words w_i and w_j .

7. Results

We compute the average topic coherence score over all the WordNet synsets using both the thesauri generated from dependency parser output and universal sketch grammar, and compare coherence scores of each other to evaluate word sketches. The higher the coherence, the better are the word sketches. Our assumption is that wordnet synsets are highly coherent. Table 1 displays the results of topic coherence over synsets at a distance of 0, 1 and 2.

From the results we observe that topic coherence of nouns and verbs at synset level is higher for thesaurus from dependency parser. This gives us an idea that word sketches of noun and verb from dependency output are more informative/accurate than from universal sketch grammar. As the distance increases, the coherence score of verbs is consistently higher for dependency parser based word sketches. This shows that dependency parser is good at capturing

verb's properties. For nouns, it is unclear why the coherence score from dependency parser is lower than universal sketch grammar at a distance of one.

For adjectives, interestingly, universal sketch grammar perform better. In our analysis we found the reason perhaps could be due to conjunction. The dependency parser always mark the conjunct word as the word in relation with target word, e.g. in the phrase *sarı/yellow ve/and kırmızı/red*, *kırmızı* is in relation *conjunction* with *ve*, resulting in the tuple (*ve*, *conj*, *kırmızı*) instead of (*sarı*, *conj*, *kırmızı*). The universal sketch grammar generates the latter tuple. A new grammatical rule which can generate the latter tuple can be written using trinary relations in Sketch Engine but we leave this work for future.

As the distance increases i.e. as the topic becomes generalized, the topic coherence is expected to decrease. But at some cases we find there is an increase in topic coherence. This might be due to fine grained classification of WordNet synsets.

Overall the results suggest that dependency parser based word sketches of nouns and verbs are relatively accurate and informative than universal sketch grammar. It is the opposite case for adjectives. We leave a thorough study on these differences for future when we have adequate resources.

8. Summary

We collected and cleaned a corpus for Turkish. We identified leading NLP tools for Turkish and applied them to the corpus. We loaded the corpus into the Sketch Engine and developed a new module that allows us to prepare word sketches directly from CONLL-format output. In addition, we presented universal sketch grammar which is language independent grammar. We generated two different thesauri from these word sketches.

We evaluated dependency parser based word sketches with universal sketch grammar by evaluating them on an external task of evaluation, the topic coherence using Turkish WordNet synsets and the thesauri generated from word sketches. Our results show that both the dependency parser based sketches are more accurate for verbs and nouns than simple sketch grammar.

In the future, we aim to build word sketches from our recent large (more than a billion size) corpora of Turkish (Baisa and Suchomel, 2012) and other Turkic languages. We anticipate that word sketches and thesauri will be of interest to linguists, lexicographers, translators, and others working closely with, or studying, the Turkish language. These word sketches are currently available in Sketch Engine.

9. Acknowledgements

We would like to thank Gülsen Cebiroğlu Eryiğit and Kenal Oflazer for their kind help on providing Turkish tools. We would also like to thank the reviewers for their suggestions on improving this work.

References

- Baisa, V. and Suchomel, V. (2012). Large corpora for turkic languages and unsupervised morphological analysis. In *Proceedings of the Eighth conference on International Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).
- Curran, J. (2004). *From Distributional to Semantic Similarity*. PhD thesis, Institute for Communicating and Collaborative Systems, School of Informatics, University of Edinburgh.
- Eryiğit, G., Nivre, J., and Oflazer, K. (2008). Dependency parsing of turkish. *Comput. Linguist.*, 34(3):357–389.
- Hudson, R. (1984). *Word Grammar*. Basil Blackwell, 108 Cowley Rd, Oxford, OX4 1JF.
- Kilgarriff, A., Kovar, V., Krek, S., Srđanović, I., and Tiberius, C. (2010a). A quantitative evaluation of word sketches. In *Proceedings of the XIV Euralex International Congress*, Leeuwarden : Fryske Academy.
- Kilgarriff, A., Reddy, S., Pomikálek, J., and PVS, A. (2010b). A corpus factory for many languages. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta.
- Kilgarriff, A., Rychly, P., Smrz, P., and Tugwell, D. (2004). The Sketch Engine. In *Proceedings of EURALEX*, pages 105–116, Lorient, France, July.
- Lin, D. (1998). Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics - Volume 2*, ACL '98, pages 768–774, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Newman, D., Lau, J. H., Grieser, K., and Baldwin, T. (2010). Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108, Los Angeles, California. Association for Computational Linguistics.
- Nivre, J. and Hall, J. (2005). Maltparser: A language-independent system for data-driven dependency parsing. In *In Proc. of the Fourth Workshop on Treebanks and Linguistic Theories*, pages 13–95.
- Oflazer, K. (1994). Two-level description of turkish morphology. *Literary and Linguistic Computing*, 9(2):137–148.
- Pomíkálek, J. (2011). *Removing Boilerplate and Duplicate Content from Web Corpora*. PhD thesis, Masaryk University.

Reddy, S., Klapaftis, I., McCarthy, D., and Manandhar, S. (2011). Dynamic and static prototype vectors for semantic composition. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 705–713, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

Rundell, M. (2002). *Macmillan English Dictionary for Advanced Learners*. Macmillan Education.

Rychlý, P. (2008). A lexicographer-friendly association score. In *Proceedings of Recent Advances in Slavonic Natural Language Processing*, RASLAN 2008, pages 6–9.

Rychlý, P. and Kilgarriff, A. (2007). An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 41–44, Stroudsburg, PA, USA. Association for Computational Linguistics.

Shieber, S. (1985). Evidence against the context-freeness of natural language. *Linguistics and Philosophy*, 8(3):333–343.

Yuret, D. and Ture, F. (2006). Learning morphological disambiguation rules for turkish. In *NAACL*, pages 328–334.