3rd International Conference on Computer Processing
in Turkic Languages (TURKLANG 2015)

Turkic language support in Sketch Engine

Vít Baisa[a,b], *Vít Suchomel*[a,b]

a NLP Centre, Masaryk University, Brno, Czech Republic
b Lexical Computing Ltd, Brighton, UK

Abstract

Sketch Engine is a corpus manager tool which allows building own text corpora from user-uploaded files or from Internet by downloading and cleaning web pages in a particular language and domain. It also provides many functions to explore the corpus data. We present the level of current support of Turkic languages (namely Azeri, Kazakh, Kyrgyz, Tatar, Turkish, Turkmen, Urdu and Uzbek) in Sketch Engine. It is currently possible to use features of Sketch Engine like concordancing, filtering, sampling, sorting of query searches, wordlist generating, collocation lists extraction, keyword extraction, finding good dictionary examples for words and phrases and some other features.

Additionally, we discuss possible developments for improving Turkic language support in Sketch Engine, starting with incorporating existing tagging tools for Turkic languages, adding terminology extraction and building word sketches and thesauri.We invite Turkic language specialists to join us in our efforts of building large scale and at the same time high quality resources for Turkic languages.

Keywords: Sketch Engine; Turkic languages; concordance; terminology extraction; word sketch; thesaurus; corpus building; language support, corpus manager

1. Introduction

Turkic language family contains more than thirty languages and the biggest language, Turkish, is spoken by almost 1% of the world population. It is spoken more than Italian or Dutch. Turkish Wikipedia is 10th (Azeri being 39th) biggest measured by the number of active editors[1] however Turkic languages in general are under-resourced from the point of view of corpus linguistics. That is why we have put some

[1] http://wikistats.wmflabs.org/display.php?t=wp&s=ausers_desc

effort to creating Turkic language resources and adding at least basic Turkic language support to Sketch Engine. This paper describes the result.

First we describe how we have built several Turkic corpora from Internet. Then we describe current features of Sketch Engine available for these corpora. A discussion of possible other usages follows. At the end we propose possible improvements and future work towards full support of Turkic languages in Sketch Engine.

2. Building Turkic corpora

We selected Turkish, Azerbaijani, Uzbek, Kazakh, Turkmen and Kyrgyz for our 2012 Turkic data collection (Baisa, Suchomel, 2012). The procedure for building general corpora from the web remains the same:

- Start with a small corpus in the target language or create one from Wikipedia texts to build language and encoding detection models.
- Find at least 100 web pages in the target language and use them as starting points for a web crawler.
- Run the web crawler – we have been successfully using SpiderLing, a text corpus oriented crawler (Suchomel, Pomikálek, 2012).
- Alternatively, run WebBootCaT (Baroni et al., 2006), a tool for creating mid-sized corpora from the web using a search engine (the tool is built in the Sketh Engine corpus creation interface) and a part of the Corpus Factory method (Kilgarriff et al., 2010).
- Clean the web data using a set of tools for HTML boilerplate removal, de-duplication (removal of similar sentences, paragraphs or documents) and a robust (web texts aware) tokenizer[2].
- Carry out part-of-speech tagging using a tagger for the target language.
- Store and index the corpus by a corpus manager to allow fast search.

Since a productive inflectional and derivational agglutinative morphology is essential for Turkic languages, any serious corpus based research can benefit from a proper morphological annotation. Although there is not a morphological analyzer built in Sketch Engine, uploading user annotated texts is supported.

Texts in languages written in multiple scripts or spoken in areas of different countries like Tatar and Uyghur are much harder to obtain using the web crawling method. Differences in alphabets and lists of words might be exploited to separate documents in different Turkic languages. Yet, one has to deal with multiple writing systems in the region: Cyrillic, Latin and Arabic.

[2] All available for free at http://corpus.tools

In case of problems stemming from the issues with crawling, we recommend the search engine driven approach to build large corpora from the web:

- Again, start with a small corpus in the target language or create one from Wikipedia texts.
- Produce a list of words in the corpus sorted by number of occurrences in the corpus from the most frequent word. Use medium frequent words, e.g. from rank 500 to 600 and from rank 1500 to 1600 as seed words for WebBootCaT.
- Let a search engine find web documents in the target language and build the corpus semi-automatically using WebBootCaT.

To gather good quality texts[3] in languages with a scarce Internet presence (which is the case of the most Turkic languages), one can employ less automated means as was shown by (Dovudov et al., 2011):

- Identify Internet sources yielding quality documents, e.g. online newspapers, and government or municipality portals.
- Analyze the web structure of the sources, i.e. locate texts within the site (e.g. find archive of a news site) and determine the important blocks in html pages: this can be automated (Song et al., 2004).
- Write a computer program downloading texts from the web according to findings in the previous step. A recursive run of wget[4] might do the task as well.

Normalization (or unification) of web texts might be required to achieve a good level of quality as reported by (Dovudov et al., 2011):

- Transliteration of letters to the desired script, e.g. from the Latin script or the Arabic script to the Cyrillic script.
- Identification and correction of language specific letters, e.g. replace H by Ң where appropriate in Kazakh, Kyrgyz, Tatar and Turkmen.

[3] A “good quality” text for the purpose of a linguistic research carried on text corpora can be defined as a long sequence of paragraphs of fluent natural sentences.

[4] Wget, a utility for downloading web content, http://www.gnu.org/software/wget/

Table 1. Turkic corpora for language research currently available in Sketch Engine

Language	Name	Corpus size [M tokens]	Lexicon size [M words]	Notes
Azeri	Turkic web – Azerbaijani	115	1.5	Web crawled
Kazakh	Turkic web – Kazakh	175	2.2	Web crawled
Kyrgyz	Turkic web – Kyrgyz	24	0.6	Web crawled
Tatar	Tatar sample	0.29	0.07	Small web corpus gathered using WebBootCaT (Ambati et al., 2012)
Turkmen	Turkic web – Turkmen	3	0.2	Web crawled
Turkish	Turkish WaC	41	1.5	Small web corpus gathered using the Corpus Factory method, parsed with MaltParser[5] (Ambati et al., 2012)
	TrTenTen	4,125	17.2	Web crawled
	OPUS2 Turkish	207	1.5	Parallel corpus[6]
Uzbek	Turkic web – Uzbek	25	0.6	Web crawled

The corpora don’t have rich metadata, e.g. domains and text types are missing for all documents. To understand the type of texts in these corpora, it is good to look at the most exploited web domains. In Table 2 you can see top domains for the Turkic corpora.

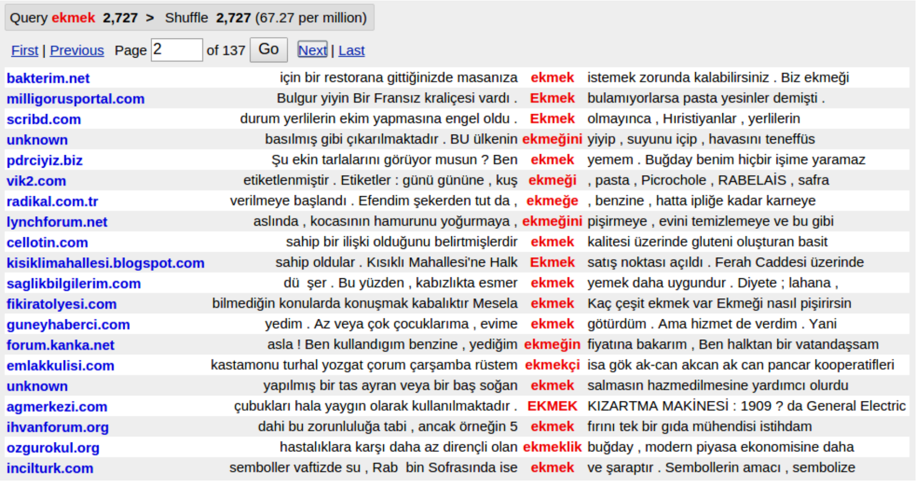
Table 2. Top domain contained in Turkic corpora

Corpus	Top domains
Turkic web – Azerbaijani	mediaforum.az, az.trend.az, milli.az, mia.az, modern.az, 525.az, ...
TrTenTen	afyonkarahisar.com.tr, savaskarsitlari.org, yeniasya.com.tr, ...
Turkic web – Kazakh	alashainasy.kz, egemen.kz, inform.kz, kaz.gazeta.kz, thenews.kz, ...
Turkic web – Kyrgyz	kabar.kg, www.azattyk.org, kg.zpress.kg, erkintoo.kg, ktrk.kg, ...
Turkic web – Turkmen	tmolympiad.org, www.azathabar.org, turkmenistan.gov.tm, cci.gov.tm, ...
Turkic web – Uzbek	uza.uz, shou-biznes.uz, jamiyatgzt.uz, old.uzbekistonovozi.uz, ...

[5] MaltParser, a data driven dependency parser. http://www.maltparser.org

[6] OPUS, the open parallel corpus. http://opus.lingfil.uu.se

3. Concordances



Query ekmek 2,727 > Shuffle 2,727 (67.27 per million)

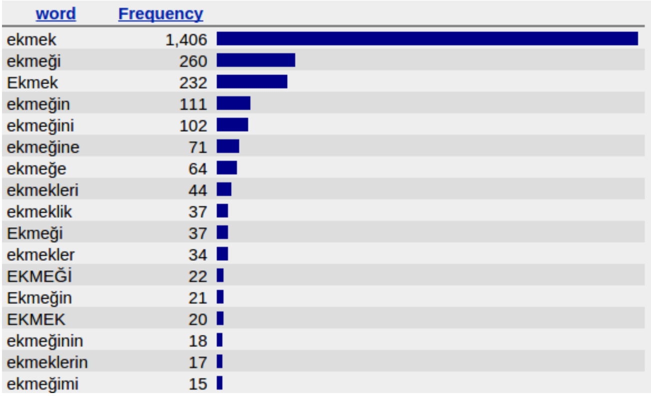
bakterim.net	için bir restorana gittiğinizde masanıza	ekmek	istemek zorunda kalabilirsiniz . Biz ekmeği
milligorusportal.com	Bulgur yiyin Bir Fransız kraliçesi vardı .	Ekmek	bulamıyorlarsa pasta yesinler demişti .
scribd.com	durum yerlilerin ekim yapmasına engel oldu .	Ekmek	olmayınca , Hıristiyanlar , yerlilerin
unknown	basılmış gibi çıkarılmaktadır . BU ülkenin	ekmeğini	yiyip , suyunu içip , havasını teneffüs
pdrciyiz.biz	Şu ekin tarlalarını görüyor musun ? Ben	ekmek	yemem . Buğday benim hiçbir işime yaramaz
vik2.com	etiketlenmiştir . Etiketler : günü gününe , kuş	ekmeği	, pasta , Picrochole , RABELAİS , safra
radikal.com.tr	verilmeye başlandı . Efendim şekerden tut da ,	ekmeğe	, benzine , hatta ipliğe kadar karneye
lynchforum.net	aslında , kocasının hamurunu yoğurmaya ,	ekmeğini	pişirmeye , evini temizlemeye ve bu gibi
cellotin.com	sahip bir ilişki olduğunu belirtmişlerdir	ekmek	kalitesi üzerinde gluteni oluşturan basit
kisiklimahallesi.blogspot.com	sahip oldular . Kısıklı Mahallesi'ne Halk	Ekmek	satış noktası açıldı . Ferah Caddesi üzerinde
saglikbilgilerim.com	dü şer . Bu yüzden , kabızlıkta esmer	ekmek	yemek daha uygundur . Diyete ; lahana ,
fikiratolyesi.com	bilmediğin konularda konuşmak kabalıktır Mesela	ekmek	Kaç çeşit ekmek var Ekmeği nasıl pişirirsin
guneyhaberci.com	yedim . Az veya çok çocuklarıma , evime	ekmek	götürdüm . Ama hizmet de verdim . Yani
forum.kanka.net	asla ! Ben kullandıgım benzine , yediğim	ekmeğin	fiyatına bakarım , Ben halktan bir vatandaşsam
emlakkulisi.com	kastamonu turhal yozgat çorum çarşamba rüstem	ekmekçi	isa gök ak-can akcan ak can pancar kooperatifleri
unknown	yapılmış bir tas ayran veya bir baş soğan	ekmek	salmasın hazmedilmesine yardımcı olurdu
agmerkezi.com	çubukları hala yaygın olarak kullanılmaktadır .	EKMEK	KIZARTMA MAKİNESİ : 1909 ? da General Electric
ihvanforum.org	dahi bu zorunluluğa tabi , ancak örneğin 5	ekmek	fırını tek bir gıda mühendisi istihdam
ozgurokul.org	hastalıklara karşı daha az dirençli olan	ekmeklik	buğday , modern piyasa ekonomisine daha
incilturk.com	semboller vaftizde su , Rab bin Sofrasında ise	ekmek	ve şaraptır . Sembollerin amacı , sembolize

Figure 1. Sampled concordance for lemma “ekmek” in TurkishWaC

The main feature available for all corpora is *concordance search*: a powerful full-text search. As many of our Turkic corpora have only word forms (lemmas and other tags are not available), the searching is limited to regular expressions over these word forms. But even with this limitation, the query language (CQL, Corpus Query Language[7]) is expressive enough to allow complex searches.

Once a result is shown, it can be sorted, further filtered (by other CQL queries), randomly sampled (see Figure 1), stored and various frequencies (Figure 2) and visualizations (Figure 3) can be obtained. All these actions can be combined to narrow and fine-tune the original result.

[7] http://www.sketchengine.co.uk/documentation/wiki/SkE/CorpusQuerying



word	Frequency
ekmek	1,406
ekmeği	260
Ekmek	232
ekmeğin	111
ekmeğini	102
ekmeğine	71
ekmeğe	64
ekmekleri	44
ekmeklik	37
Ekmeği	37
ekmekler	34
EKMEĞİ	22
Ekmeğin	21
EKMEK	20
ekmeğinin	18
ekmeklerin	17
ekmeğimi	15

Figure 2. The most frequent wordforms of “ekmek” in TurkishWaC

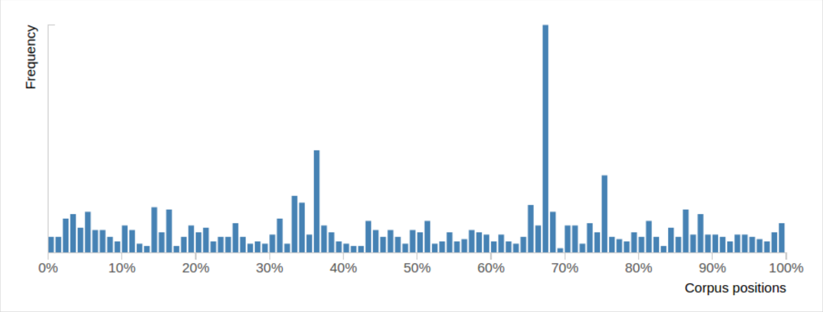
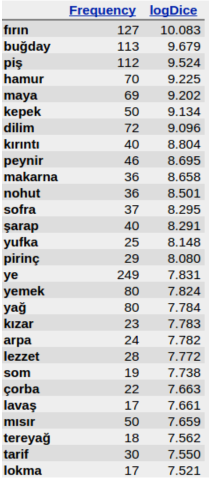
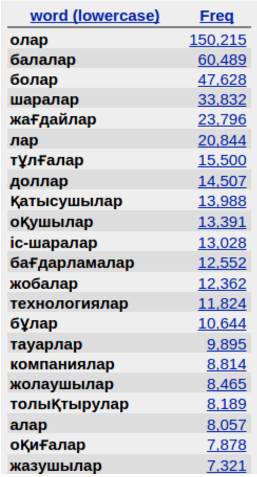


Figure 3. Frequency distribution of lemma “ekmek” in corpus parts

If there are enough hits (examples) in a concordance search, one can extract the most salient collocations from it. The algorithm in Sketch Engine looks for the most frequent words which co-occur with the searched query and then applies a co-occurrence statistics. We usually use *logDice* (Rychlý, 2008). In Figure 4a you can see collocates derived from the concordance for “ekmek”.



	Frequency	logDice
fırın	127	10.083
buğday	113	9.679
piş	112	9.524
hamur	70	9.225
maya	69	9.202
kepek	50	9.134
dilim	72	9.096
kırıntı	40	8.804
peynir	46	8.695
makarna	36	8.658
nohut	36	8.501
sofra	37	8.295
şarap	40	8.291
yufka	25	8.148
pirinç	29	8.080
ye	249	7.831
yemek	80	7.824
yağ	80	7.784
kızar	23	7.783
arpa	24	7.782
lezzet	28	7.772
som	19	7.738
çorba	22	7.663
lavaş	17	7.661
mısır	50	7.659
tereyağ	18	7.562
tarif	30	7.550
lokma	17	7.521



word (lowercase)	Freq
олар	150,215
балалар	60,489
болар	47,628
шаралар	33,832
жаҒдайлар	23,796
лар	20,844
тҰлҒалар	15,500
доллар	14,507
Қатысушылар	13,988
оҚушылар	13,391
іс-шаралар	13,028
баҒдарламалар	12,552
жобалар	12,362
технологиялар	11,824
бҰлар	10,644
тауарлар	9,895
компаниялар	8,814
жолаушылар	8,465
толыҚтырулар	8,189
алар	8,057
оҚиҒалар	7,878
жазушылар	7,321

Figure 4. (a) Collocations for lemma “ekmek” *(b)* Kazakh wordlists for words ending *with “лар”*

4. Wordlists

Wordlist is another feature universally available for any corpus. Any positional attribute (word, lemma, part of speech, morphology tag, …) can be explored. It is similar to frequency lists of concordance search but wordlists are more general. E.g. you can get the most frequent words, the most frequent lemmas ending with “лар” etc. You may use several constraints and filter the results with regular expressions. You can obtain either raw frequencies or document frequencies per item. In Figure 4b you can see wordlist for all words ending with “лар” from the Kazakh corpus.

5. Word sketches and thesaurus

Word sketches are the core feature of Sketch Engine (hence the name). Word sketch is a one-page, automatic, corpus-derived summary of a word’s grammatical and collocational behaviour. There are several ways to build word sketches.

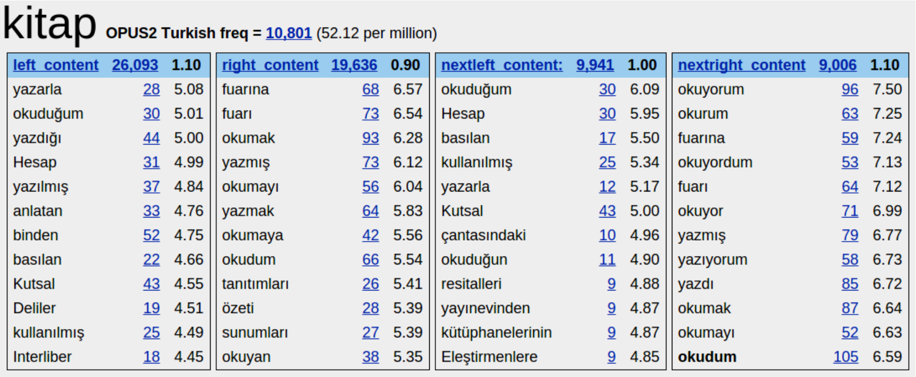
5.1 CoNLL

We have developed a script which takes CoNLL-annotated corpus as input and generates word sketch grammars[8]. This was also applied on Turkish (Ambati et al., 2012).

[8] http://www.sketchengine.co.uk/documentation/wiki/SkE/SketchesFromCONLL

5.2 *Universal word sketch grammar*

We have also processed Turkish part of OPUS parallel corpus using rudimentary tagging (content words, punctuation, numbers) together with so called universal word sketch grammar with very simple rules like “content word to the left from a headword” and other analogous rules. This processing yielded word sketches which can be built also for other Turkic languages but which are not of very high quality and usability. See Figure 5.



kitap OPUS2 Turkish freq = 10,801 (52.12 per million)

left_content	26,093	1.10	right_content	19,636	0.90	nextleft_content:	9,941	1.00	nextright_content	9,006	1.10
yazarla	28	5.08	fuarına	68	6.57	okuduğum	30	6.09	okuyorum	96	7.50
okuduğum	30	5.01	fuarı	73	6.54	Hesap	30	5.95	okurum	63	7.25
yazdığı	44	5.00	okumak	93	6.28	basılan	17	5.50	fuarına	59	7.24
Hesap	31	4.99	yazmış	73	6.12	kullanılmış	25	5.34	okuyordum	53	7.13
yazılmış	37	4.84	okumayı	56	6.04	yazarla	12	5.17	fuarı	64	7.12
anlatan	33	4.76	yazmak	64	5.83	Kutsal	43	5.00	okuyor	71	6.99
binden	52	4.75	okumaya	42	5.56	çantasındaki	10	4.96	yazmış	79	6.77
basılan	22	4.66	okudum	66	5.54	okuduğun	11	4.90	yazıyorum	58	6.73
Kutsal	43	4.55	tanıtımları	26	5.41	resitalleri	9	4.88	yazdı	85	6.72
Deliler	19	4.51	özeti	28	5.39	yayınevinden	9	4.87	okumak	87	6.64
kullanılmış	25	4.49	sunumları	27	5.39	kütüphanelerinin	9	4.87	okumayı	52	6.63
Interliber	18	4.45	okuyan	38	5.35	Eleştirmenlere	9	4.85	**okudum**	105	6.59

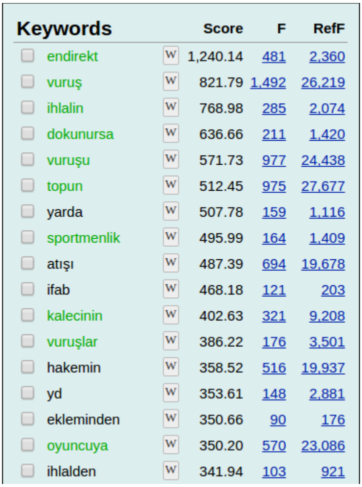
Figure 5. Universal word sketch for “kitap” in OPUS corpus.

5.3 *Word sketch grammar*

The last and the most advanced way is to write grammar rules manually. It needs both tagged corpus and a language specialist. This is yet to be done.

6. Keyword extraction

If you build your own domain-specific corpus, you can extract keywords from it. The extraction procedure depends on relative frequencies of words in your corpus and in a reference corpus in the same language. For the purpose of this paper we have built a small Turkish corpus using football seed words (a few terms from *Futbal* article on Turkish Wikipedia). Several pages were automatically downloaded and then the corpus was expanded a little with WebBootCat tool, yielding cca 250,000 tokens from football-related Internet pages in Turkish. In Figure 6 you can see the top part of the resulting list of keyword candidates from the domain corpus.



Keywords			Score	F	RefF
□	endirekt	W	1,240.14	481	2,360
□	vuruş	W	821.79	1,492	26,219
□	ihlalin	W	768.98	285	2,074
□	dokunursa	W	636.66	211	1,420
□	vuruşu	W	571.73	977	24,438
□	topun	W	512.45	975	27,677
□	yarda	W	507.78	159	1,116
□	sportmenlik	W	495.99	164	1,409
□	atışı	W	487.39	694	19,678
□	ifab	W	468.18	121	203
□	kalecinin	W	402.63	321	9,208
□	vuruşlar	W	386.22	176	3,501
□	hakemin	W	358.52	516	19,937
□	yd	W	353.61	148	2,881
□	ekleminden	W	350.66	90	176
□	oyuncuya	W	350.20	570	23,086
□	ihlalden	W	341.94	103	921

Figure 6. Keyword extraction from a domain-specific (football) corpus

The green keywords were used in building the corpus with WebBootCat. Sketch Engine shows also links to related Wikipedia articles (Turkish Wikipedia in this case). The score expresses how salient a keyword is in the domain corpus when compared with a general (much bigger) Turkish reference corpus. The last two columns are raw frequencies in the focus and in the reference corpus. It is also important to note that neither of the authors has any knowledge of Turkish language thus it is possible that the keywords are not perfect. The same methods could be used to build e.g. Tatar corpus and extract keywords from it as it is fully statistically-based approach. More info about the extraction procedure can be found in (Kilgarriff, 2014).

7. Further work and development

The support for Turkic language can be substantially improved. The two most beneficial improvements are discussed below.

7.1 Term extraction

Recently we have developed term extraction for several languages: English, Spanish, German, Czech and a few others (Kilgarriff, 2014). To add a new language to the list, it is necessary to describe possible terms (usually noun phrases) using advanced CQL queries. These queries both describe the grammar rules for matching all possigle term phrases but also they describe how the resulting basic word form for terms should look like.

7.2 Morphological analyzer integration

Advanced Sketch Engine features, such as word sketches and thesaurus, or querying the corpus for morphological categories require a morphologically annotated corpus. Although annotated texts can be loaded into the Sketch Engine, it would be much more convenient for anyone building a Turkic corpus if the tool made the tagging for them.

The requirements for embedding a morphological analyzer in the corpus building interface are:

- Software running in a Unix-like environment.
- Command line interface for batch processing of large quantities of data.
- Documentation: evaluation of the tagger, description of possible output tags.
- Licence allowing to incorporate the tool in Sketch Engine.

8. Conclusion

We have described the current support of Turkic languages in Sketch Engine. It enables a basic analysis and users can upload preprocessed data and use all the standard features of Sketch Engine. With this paper we hope to attract Turkic language specialists to use this powerful tool for exploring the richness of all Turkic languages. Sketch Engine is currently used at many language institutions in Europe and we think that it can boost language research of Turkic languages, its lexicography, terminology and linguistics in general.

Acknowledgements

This work has been partly supported by the Ministry of Education of CR within the LINDAT-Clarin project LM2010013.

References

Baisa, V., 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).*

Suchomel, V., Pomikálek, J. (2012). *Efficient web crawling for large text corpora*. In *Proceedings of the seventh Web as Corpus Workshop (WAC7).* pp. 39-43.

Baroni, M., Kilgarriff, A., Pomikálek, J., Rychlý, P. (2006). *WebBootCaT: a web tool for instant corpora.* In *Proceeding of the EuraLex Conference*, pp. 123-132.

Dovudov, G., Pomikálek, J., Suchomel, V., Šmerk, P. (2011). *Building a 50M Corpus of Tajik Language.* In *RASLAN 2011 Recent Advances in Slavonic Natural Language Processing.*

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

Song, R., Haifeng L., Ji-Rong W., Wei-Ying M. (2004). *Learning block importance models for web pages.* In *Proceedings of the 13th international conference on World Wide Web*, pp. 203-211. ACM.

Kilgarriff, A., Reddy, S., Pomikálek, J., Avinesh, P. V. S. (2010). *A Corpus Factory for Many Languages.* In *LREC 2010*.

Kilgarriff, A., Jakubíček, M., Kovář, V., Rychlý, P., Suchomel, V. (2014). *Finding terms in corpora for many languages with the Sketch Engine*. EACL 2014, 53.

Ambati, B. R., Reddy, S., Kilgarriff, A. (2012). *Word Sketches for Turkish*. In *LREC*. pp. 2945-2950.

Kilgarriff, A., Rychlý, P., Smrž, P., Tugwell, D. (2004). *The Sketch Engine*. Information Technology. 2004.