Hungarian-Somali-English online dictionary and taxonomy

István Endrédy
Pázmány Péter Catholic University, Faculty of Information Technology and Bionics
50/a Práter Street, 1083 Budapest, Hungary

MTA-PPKE Hungarian Language Technology Research Group
50/a Práter Street, 1083 Budapest, Hungary
istvan.endredy@gmail.com

Abstract

Background. The number of Somalis coming to Europe has increased substantially in recent years. Most of them do not speak any foreign language, only Somali, but a few of them speak English as well.
Aims. A simple and useful online dictionary would help Somalis in everyday life. It should be online (with easy access from anywhere) and it has to handle billions of word forms, as Hungarian is heavily agglutinative. It should handle typos as the users are not advanced speakers of the foreign languages of the dictionary. It should pronounce words, as these languages have different phonetic sets. It should be fast with good precision because users do not like to wait. And last but not least, it should support an overview of the vocabulary of a given topic.
Method. A vocabulary (2000 entries) and a taxonomy (200 nodes) was created by a team (an editor and a native Somali speaker) in an Excel table. This content was converted into a relational database (*mysql*), and it got an online user interface based on *php* and *jqueryui*. Stemmer and text-to-speech modules were included and implemented as a web service. Typos were handled with query extension.
Results. Although the dictionary lookup process does stemming with a web service and makes a query extension process, it is very fast (100-300ms per query). It can pronounce every Hungarian word and expression owing to the text-to-speech web service.
Conclusion. This dictionary was opened to the public in October, 2013. (*http://qaamuus.rmk.hu/en*) The next step is the creation of a user interface optimised for mobile devices.

Keywords: online dictionary, taxonomy, Somali

1. Introduction

In the past years, the number of immigrant Somalis in Hungary has increased. Most of them do not speak any language except for Somali, but a few of them speak English, too. They can not manage their business without being able to communicate effectively, so they need local help.
Some of the immigrants asked for help at the Reformed Mission Center (*Református Missziói Központ*), where they got the opportunity to learn Hungarian as a foreign language. This helps them a lot in becoming independent.
Somali dictionaries are not easily accessible, especially not in the Hungarian–Somali direction. Therefore an online Somali dictionary was developed that can be used almost from everywhere. This project was started in the framework of the School Integration Programme of the Refugee Mission, funded by the European Refugee Fund (*Menekültmisszió Iskolai Integrációs Programja, Európai Menekültügyi Alap*).
László Joachim (a native Hungarian) created a Hungarian-Somali-English dictionary in the form of an Excel spreadsheet with the help of a native Somali speaker, Tukale Hussein Muhyadin. The dictionary contained a basic vocabulary of about 2000 entries, and a taxonomy that had 200 nodes. This database served as a basis for the online dictionary.

2. Available solutions

There are very few online Somali-English dictionaries[1]. They do not use stemming and they cannot correct spelling errors on input. At the time of development there was only one Somali–Hungarian dictionary[2] on the web, which was a community built word list with 865 translations.
The situation changed on 10th December 2013, when Google introduced[3] Somali on its popular Google Translate service. Although this application can even translate full sentences, erroneous translations are very frequent (Table 1).
Google Translate is based on statistical machine translation. In theory, the more example sentence pairs there are in the training corpus of translation system, the better the translations are. The system may improve in time, but if a language is highly agglutinative, this method can not learn every possible phrase and sentence. Hungarian words have many forms. Nouns might have several thousands word forms, verbs might have thousands of different word forms (cf. Section 4 below). Owing to the practically infinite number of possible word forms and the relatively free word order of Hungarian, the quality of conventional statistical machine translation for Hungarian will never be perfect. For example, Hungarian *elmehettetek* 'you may have left' is unknown for Google Translate, it is an infected word form of *elmegy* (last row of Table 1). Futhermore, Somali is also heavily agglutinative. This causes even more difficulty in translation.
Google usually translates between different languages through English. For example, it first translates from Somali to English, then from English to Hungarian. These

[1] http://www.afmaal.com/dictionary, http://www.freelang.net/online/somali.php?lg=gb

[2] http://en.glosbe.com/hu/so

[3] http://www.webpronews.com/google-translate-hits-80-languages-milestone-adds-9-new-ones-2013-12

input words	Google English	Google Somali	Google Hungarian	Our English	Our Somali	Our Hungarian
big		weyn ✓	nagy ✓		wayn ✓	nagy ✓
nice		- ✗	szép ✓		macaan ✓	finom, kedves ✓
went		u galay ✗	ment ✓		tagid ✓	megy ✓
high		sare ✓	nagy ✗		dheer ✓	magas ✓
degaan	residence ✓		tartózkodás ✓	accommodation ✓		szállás ✓
isku eeg	to see ✗		hogy ✗	similar ✓		hasonló ✓
jön	come ✓	yimaado ✓		come ✓	kaalay ✓	
nagy	high ✗	sare ✗		big, large ✓	weyn ✓	
elmehettetek	- ✗	- ✗		leave ✓	tagid ✓	

Table 1: Google translate test 17/12/2013

steps may include errors, a single error (in any of the steps) may impair the final translation. This type of error can be seen in Table 1, at the input word *nagy*. Google translates this Hungarian word as *high*, but this is not correct: it means *big, large*. This error occurs in the Hungarian–Somali direction as well: *nagy* is translated to Somali *sare* just like English *high*. As we can see Google uses English as intermediate language between Hungarian and Somali. This solution may impair the quality of translation.

To sum it up, Google Translate has errors in Somali-Hungarian translations according to tests. It is due to the fact that both languages are agglutinative, and GT translates with English as an intermediate language. A small mistake at any level may result finally in a poor translation. It should be used carefully in case of these languages. Consequently, a dictionary tool is needed for accurate translation between Somali-Hungarian-English instead of GT.

3. Architecture and modules

This project aimed to create an online dictionary for Somali immigrants. The content of the dictionary was imported into a relational database (*mysql*), with a few tables (details in Figure 3). The web interface was developed in *php* and *jqueryui*. The database design, data migration and the development of the web interface were done in this project. English and Hungarian stemmer and text-to-speech modules were used out of the box as a web service. The stemmer we used in this project is based on the morphological analyzer engine HUMOR ('High speed Unification MORphology') developed at MorphoLogic (Prószéky and Kis, 1999). The stemmer was implemented by the author. The TTS engine we used is Profivox (Olaszy et al., 2000) with Microsoft Speech API. The TTS and the stemmer service is provided by *morphologic.hu*. The basic dataflow of the system is illustrated on Figure 1.

4. Content of the dictionary

Size and structure. There are 2,000 entries, and 200 taxonomy nodes in the dictionary. Each entry has several fields, as shown in Table 2.

The Taxonomy field contains nodes which are related to the entry. For instance, "vegetable" belongs to the "food" and "vegetables" groups. These connections help students to explore or refresh the vocabulary of a given topic, by listing the child nodes of the taxonomy.

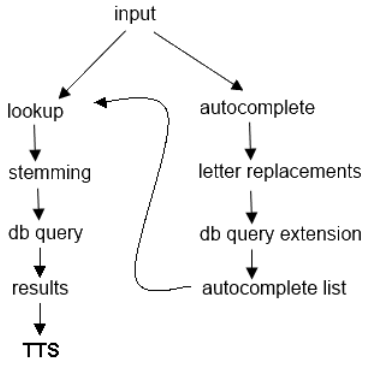

Figure 1: Dataflow of the system

field name	example value
English	would you mind, if...?
Hungarian	*baj, ha...?*
Hungarian keyword	*baj*
Somali	dhib male, hadii
part of speech (keyword)	noun
pronunciation	
other forms, grammatical information	
usage	
examples	Hu: *Nem baj, ha kinyitom az ajtót?* So: Dhib malah, hadaan daqada furo? En: Would you mind if I opened the door?
taxonomy	So: qalab wax sahlaya/karaan, awood, mug/ogolaansho, rukhsad Hu: *lehetőség/ képesség/ engedély* En: opportunity/ ability/ permission

Table 2: Example entry from the Excel table of the dictionary

Importing entries into the database. The editors of the dictionary created entries in an Excel table and a taxonomy hierarchy in a MS Word document. The entries were exported in csv (comma separated value) format. In this form they can be easily imported into a relational database, such as MySQL. The SQL table structure reflects that of

the columns of the Excel table (columns are illustrated in Table 2). The key columns have been indexed as well (Hungarian keyword, Somali, English). These columns became searchable.
The taxonomy did not have such a strict format, therefore it was parsed with a php script, and each taxonomy entity was put into a database table. The connection between words and their connections to the taxonomy were defined with the help of the "taxonomy" column of the Excel table. A word may have several taxonomy connections, it is a one-to-many relation in the database.
Some taxonomy entries were poorly formatted (missing a delimiter between different languages, or other syntax errors). In such cases, errors were corrected one by one or with the help of the editors.

An example fragment from the taxonomy Word document illustrates (Figure 2) that its format was not computer friendly. The content is bilingual without delimiters, therefore structure and content were not easy to parse (English translation is only for illustration)

1. DAD – *AZ EMBER* MAN
macluumaadka shakhsiga *személyes adatok:* personal data
1 macluumaadka shakhsiga guud ahaan *személyes adatok általában* personal data in general
2 magaca-qofka *név* name
3 ciwaan *lakcím,* address
4 da' *életkor* age

Figure 2: Example taxonomy entry

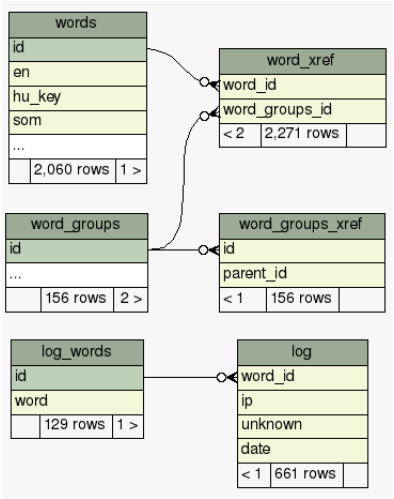

Figure 3: Relations between datatables

5. Features

The development of this online dictionary focused on the features which are important for foreign speakers. The following sections present the main features which are not available in other Somali online dictionaries.
A basic requirement of the application was to be user friendly and fast with good precision, despite the differences between the three languages (different phonetic sets, stemming rules and different types of frequent typing mistakes). The vocabulary should help Somali users to solve the most typical situations of everyday life.
Correction of typical typos. Hungarian has some digraphs and trigraphs which are difficult to write for a foreign speaker. The application replaces the typical typos with the correct word forms; otherwise the users will not find the searched word and will not learn the correct spelling. Typical typos were collected from all the three languages involved in this project, and search terms are completed with suggestions. This operation is triggered when the user types into the search input field. At this point, an autocomplete list is shown, and the user can click on a suggestion. (This process is described in detail in Section 6.)

Figure 4: The autocomplete feature with suggestions

Table 3 contains the typical letter replacements which are used for the creation of the autocomplete suggestion list.

Typical consonant replacements		
Consonants		Vowels
tsz → c	j → ly	i+<vowel > → ij+<vowel> ("fiatal" → "fijatal")
tz → c	nj → ny	e,é → i,í
dj → gy	f → v	a → e
dzs → gy	d → t ("fáratt")	a → o
cs → gy	s → sz	o → u
b → p	z → sz	fel → föl
ts → cs	sz → ssz	with and without accent: aeoiu → áéúöőüűóí
tj → ty	zs → sz	o → öőó
th → t	sz → s	u → üúű
lj → ly	sz → z	
l → ll ("szálás")		

Table 3: Letter replacements at query time

The application is capable of handling Hungarian di- and trigraphs, typical mistypings and phonetic mistakes, so it can find words in a great distance. For example for Hungarian "fijatal" the program will find "fiatal", or for "tsolad" the program will find "család". Of course these strings seem to be similar for a human, but for the computer these strings are very different. It is not trivial to find them based on these inputs.

Our application is capable of finding the correct spelling form even for strings with multiple errors.
Why does not provide the user interface a phonetized input option, a keyboard with phonetic input which may solve the problem of spelling errors and orthographic variations? In some languages, for instance French, phonetic input may help the users when a phoneme may have several letter combinations. Specifically if you do not know the spelling of 'éléphant', you can type with phonetic input 'elefan', and it will find the word correctly. In this case, Hungarian phonemes and letters are unknown for a Somali speaker, in addition Hungarian di- and trigraphs have different pronunciation (*gy, ty, ny, sz, zs, dz, dzs*, etc.). Consequently, a phonetic keyboard could not help, because the user does not know which letter or phoneme is necessary in the given word. We found it to be more comfortable for the user just to type the word, and correction is done on the fly with the autocomplete list. The Somali phonemes and letters are replaced with the possible Hungarian equivalents on each key press and the user may choose the correct form from the list.

Input stemming. Hungarian is an agglutinative language: one word (especially verbs) may have more than one thousand word forms (Oravecz and Dienes, 2002). In addition, Somalis most probably cannot type Hungarian words correctly. That is why the online user interface has to support typos and handle word stems: it has to find the entries by any word form of a given word. At query time the stem of the word is also searched in the dictionary. For example, if the user searches for *vagyok* 'I am', then its stem *van* 'is' will also be looked up. This feature increases the recall of the query results. Stemming is available in English and Hungarian as web services at *morphologic.hu*. The dictionary makes a web service call each time it needs to stem a word in these languages, and stems will be looked up in the dictionary as well. This way the user has the opportunity to copy/paste words in the form they occur in the original context, and the dictionary can find them easily.

Pronunciation: the text-to-speech module. Hungarian and Somali letter-to-sound rules differ considerably. For example, several sounds are marked by a single consonant letter in Somali while by a digraph in Hungarian. (e.g. the sound /s/ is marked by 's' in Somali, while by 'sz' in Hungarian). Due to these differences the application should be able to pronounce words as well. This application makes language learning easier. A text-to-speech module is available for Hungarian as a web service at *morphologic.hu*. If the user clicks on the icon "Listen", a web service call will be made, and the text can be listened to.

Multilanguage options. Visitors can select the language of the online user interface: it can be Somali, English or Hungarian. (The default setting is Somali since it is intended for Somali speakers.) The user can also set the language of the query.
At the beginning, the default source language of the search was Hungarian. But the first experiences showed that visitors type words in all three languages. Therefore the default setting is now to search in all three languages. The dictionary looks up words in each language, thus the 'not found' message became rarer.

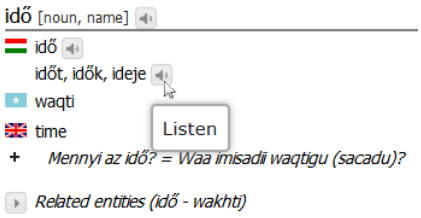

Figure 5: Text-to-speech module on the user interface

Taxomony. Our dictionary had a requirement that it should facilitate the overview of the vocabulary of a given topic. To attain this goal, we used a taxonomy. Although it would have been possible to use an existing semantic resource, we decided to create one of our own, as we found the hierarchy in the existing resources too detailed. The main consideration of the taxonomy nodes was the everyday usability from the apects of Somali immigrants.
DBpedia (Auer et al., 2007) has large scope with many nodes, but our project needs a taxonomy supporting at least two languages of the dictionary. DBpedia has English, but neither Hungarian nor Somali is included among the supported languages.
Although YAGO (Suchanek et al., 2008) has labels in Hungarian besides English, it has an extremely high granularity: several types of relations and levels, just like WordNet. YAGO covers a huge amount of concepts, people, organizations, geographical locations. For our project, only a basic subset of nodes, about 2% of the knowledge in YAGO would be needed.
Lexvo (de Melo and Weikum, 2008) has English and Hungarian translation as well. Although the taxonomy it is based on is almost a detailed as YAGO, we consider it as a potential source of extension for our taxonomy. As a first approach, Lexvo is connected to the dictionary in a light way. Each entry has a *related Lexvo taxonomy* link which may show the related nodes, translations and definitions from Lexvo. The connection is lazy: Lexvo content is downloaded on the fly based on the Hungarian keyword. Therefore an entry may show the related Lexvo nodes on the front end, with its sisters and parents. Further possibilities are discussed in Section 9 below.

Exploring the taxonomy in two ways. During this project, a taxonomy was also built, which represents topic nodes and their semantic connections. For instance, root nodes are *man*, *communication*, or *properties of things*. These nodes have child nodes; moreover each node may have connections to other nodes.
Entries of the dictionary may also have connections to these taxonomy nodes. These connections can be used to show related content. Entries with strong semantic connections can be listed or explored. There are two entry points to viewing the taxonomy: a bottom up and a top down approach.
Moreover, topic nodes can be explored in a hierarchical view, and each topic (or node) may list its children. This

way the vocabulary of a special topic can be listed. Topic-driven exploration helps language learners to look for a word or to revise the vocabulary of a semantic field fast. This function is available in a separate menu. In this case, the root nodes are shown by default (e.g. *man, things, habits*), and each node can be opened to reveal its child nodes.

On the other hand, an entry may show which other entries are connected to the same topic node. For example the *primary school* entry has a connection to the *school types* taxonomy node. Then every connected entry of *school types* taxonomy node will be shown when displaying the *primary school* entry.

Every word entry has a "related entries" link. At this point the user can view its sibling and parent nodes in the taxonomy hierarchy. This is illustrated in Figure 5. There is also a "more related entries" link, which displays the parent node of the entry. This possibility may give a bigger overview of the semantic group of the given entry.

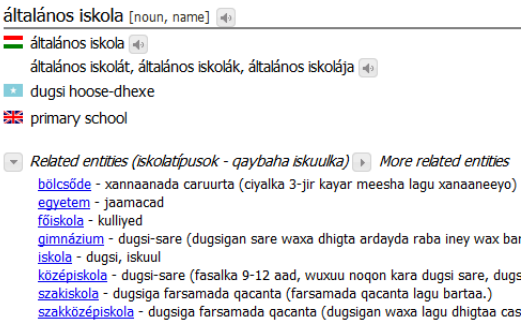

Figure 6: Related (sibling) entries

Feedback. The project had a requirement that the users should have the ability to report if a word is missing or they have any problem with the dictionary. Therefore a feedback user interface was developed, which sends an email to the administrator with the user's message.

6. Online administrative features

The content is constantly enlarged by the editors, therefore an online administration user interface was developed. It is more powerful than importing Excel tables. On the one hand, importing fails if a delimiter is missing or a new column appears: it is not a fault-tolerant process. On the other hand, online editing has the benefit that every modification is immediately ready for use by the public.

Editable entries + taxonomy. Each property of the entry can be edited. Some of them have an autocomplete feature: if the administrator starts to type in the field 'part of speech' or 'taxonomy connections', potential suggestions are displayed. This method fastens the process of editing, and it keeps these fields more consistent (see Figure 7).

There is an option to upload images or videos to an entry.

Editing can be started from the administrator's interface and from the public interface as well. (If one is logged in as an administrator, an edit icon appears next to each entry.)

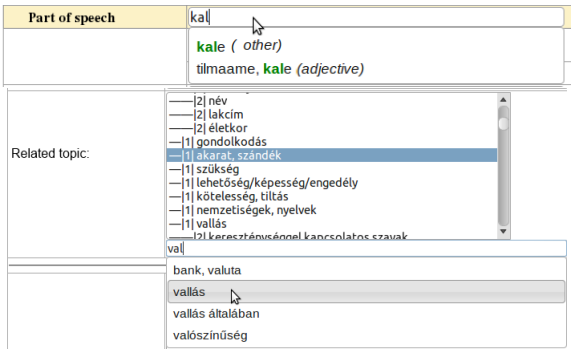

Figure 7: Autocomplete features on the admin interface

Logging queries. It is useful for editors to look at the searched words. It can answer such questions as: what is important for users, what is missing, which topic is the most popular this month, what kind of words were interesting for a user in one session?

Therefore each searched word is logged with the following pieces of information: known or unknown word, timestamp, ip address (just for identifying the user session).

Google analytics is also used independently, to analyse visitor information.

7. Example of usage

A user would like to find the meaning of the word *'young'*. He can not spell this word correctly, and types *'yuung'* into the input field. The autocomplete feature of the dictionary application suggests words for this string, in other words, it corrects the input to the forms which are known to this dictionary. This step is illustrated in Figure 8.

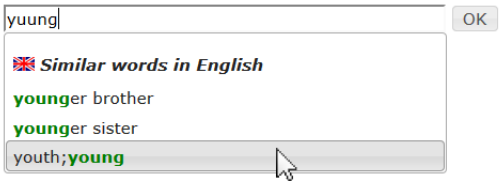

Figure 8: Autocomplete feature

Even if the user types the word correctly, the autocomplete feature makes the typing and inquiry faster. As it saves time, so users usually like it. At this point, the user may choose from the suggestion list. In this case, the intended word is in the last line (Figure 8). The search is started, and the user gets the results, the screenshot presented in Figure 9. It is important to mention that at this point (when the user clicks on an option in the autocomplete list), no automatic correction is done. The suggestion list contains only correct words, consequently it would be unnecessary to make corrections or suggestions on this input as well. If the user chooses a word from the suggestion list, the application takes the user's input as it is and looks it up without any automatic correction. Otherwise, similar entries would be noise in the result.

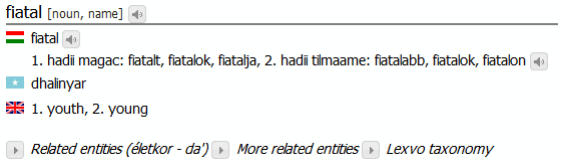

Figure 9: Example entry

8. Discussion

Other online Somali dictionaries lack autocomplete search and they do not handle typos either, and most of them have no text-to-speech option. Our solution goes beyond the earlier attempts. It is a big advantage that one can practice the pronunciation of a given word: it is practical for foreign speakers, especially for language learners. If the user can not remember the exact spelling of a word, and (s)he types a similar word (in other words: (s)he spells it incorrectly), our application will find it despite the errors. When you paste an unknown and inflected word from a text, a dictionary without stemming can not find its entry, especially in Hungarian where words may have very different forms. Our application includes stemming, so inflection is not a problem.

As for direct feedback, the editors of the dictionary are satisfied with the administrative features. Users have just started to use the application, therefore it is early to evaluate the project.

9. Future plans

The next step in the project could be the creation of a mobile application or a mobile-optimized web page. This way the dictionary could be used easily from anywhere. The dictionary service would be accessible in a comfortable way.

However, the exact improvements and changes will be based on feedback from the users, so that the program could satisfy real needs. Therefore the service will follow the requirements.

The present content of the dictionary is tuned to beginners' needs, with a basic vocabulary. The size of the vocabulary may be increased in the future. A wider entry set might serve professional needs as well.

Input stemming is done only in English and Hungarian. A Somali morphology and stemmer would increase the precision of the dictionary.

The taxonomy used in the dictionary is connected and completed with information from the Lexvo system. But this connection is created only on the fly, the related nodes are downloaded from Lexvo.org when user clicks on it. As a further step, Lexvo may be integrated in a deeper way. It can also be used as a source of additional nodes to our taxonomy and the corresponding dictionary nodes by importing English and Hungarian labels from Lexvo (possibly with manual correction in case of mistranslations) and opening up the possibility of supplying a Somali translation to users who have some knowledge of English in addition to Somali.

10. Conclusion

An online Somali-English-Hungarian dictionary was developed in this project for the Somalis who started to live in a foreign language environment (*http://qaamuus.rmk.hu/en*). The main aim was to help them in the most common situations, such as settling an administrative issue in an office, or shopping. It is important for them to be able to manage their business on their own, to live as ordinary citizens.

The features and the structure of the application were designed to serve the typical needs of language learners: assisting them in the process of learning how to write, pronounce and use words correctly. Entries were also selected for beginners, thus the vocabulary is composed of a basic vocabulary of everyday usage.

Administrators of the dictionary can edit the contents online, which is comfortable and the entries are ready for the public immediately after the modification.

Users can send feedback to the editors with a single click. This kind of direct feedback may result in a better and more usable dictionary.

11. Acknowledgements

I would like to express my gratitude to Dr Nóra Wenszky and her husband Attila Novák for their constructive and patient suggestions during the writing of this article. This work was partially supported by TÁMOP – 4.2.1.B – 11/2/KMR-2011-0002 and TÁMOP – 4.2.2/B – 10/1–2010–0014.

12. References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *Proceedings of the 6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference*, ISWC'07/ASWC'07, pages 722–735, Berlin, Heidelberg. Springer-Verlag.

de Melo, G. and Weikum, G. (2008). Language as a foundation of the Semantic Web. In Bizer, C. and Joshi, A., editors, *Proceedings of the Poster and Demonstration Session at the 7th International Semantic Web Conference (ISWC 2008)*, volume 401 of *CEUR WS*, Karlsruhe, Germany. CEUR.

Olaszy, G., Németh, G., Olaszi, P., Kiss, G., Zainkó, C., and Gordos, G. (2000). Profivox — A Hungarian text-to-speech system for telecommunications applications. *International Journal of Speech Technology*, 3(3-4):201–215.

Oravecz, C. and Dienes, P. (2002). Efficient stochastic part-of-Speech tagging for Hungarian. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC2002*, pages 710–717, Las Palmas.

Prószéky, G. and Kis, B. (1999). A unification-based approach to morpho-syntactic parsing of agglutinative and other (highly) inflectional languages. In Dale, R. and Church, K. W., editors, *ACL*. ACL.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2008). Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6(3):203–217, September.