LOTKS 2017

Workshop on Language, Ontology, Terminology and Knowledge Structures

On September 19th the second edition of the Language, Ontology, Terminology and Knowledge Structures (LOTKS) workshop will take place as a satellite workshop of the 12th International Conference on Computational Semantics (IWCS) in Montpellier, France. Following on from a successful first edition as a joint workshop at LREC 2016, the intention is once again to provide a forum for different research communities to interact and discuss issues within the intersection of computational linguistics, ontology engineering, knowledge modelling and terminologies.

LOTKS grew out of the need for a workshop that dealt, on the one hand, with enhancing knowledge bases or conceptual schemes with linguistic knowledge, as well as on the other, the growing use of ontologies and concept schemes to enrich linguistic or lexical datasets -- in particular computational lexicons.

The workshop also offers showcasing the use of conceptual/terminological/ ontological resources in NLP or computational linguistics in general. This year we have introduced new themes relating to the use of terminology schemes and ontologies in the digital humanities. The workshop welcomes contributions from both academics and industry professionals.

Fahad Khan
Istituto di Linguistica Computazionale (A. Zampolli) – CNR

https://langandonto.github.io/langonto-termiks-2017/

to place /r/ together with /l/. However, since such a split never occurs before or after s, ʃ it is not possible to place the silibants to conform to this rule.

We tested this sonority rule on our database (omitting syllables with silibants in the onset). The theory successfully predicted the omission/inclusion of /e/ in 88.3% of words.

We have developed an alternative algorithm with better performance: When *schwa* immediately follows the first letter it is pronounced /e/ if and only if at least one of the following occurs:

- The first phoneme is a word prefix, such as b (=in) v (=and).
- The first phoneme is a verb conjugation prefix, e.g., tsaˈper te.sa.ˈper, *you will tell* = t (future, 2nd person)+saˈper.
- The first phoneme is j,l,m,n,r.
- The second phoneme is ʔ,h,ʕ .
- If *schwa* occurs elsewhere it is pronounced /e/ if and only if it is:
  - The second *schwa* in the pattern C1 *schwa* C2 *schwa* C3 (Ci a consonant).
  - Between two identical or similar letters (e.g., between /d/ and /t/)

The first two rules require a morphological analyzer to identify the correct analysis of a word in context. Since we did not have at our disposal a morphological analyzer for dotted texts, we could not apply these rules, which could have prevented at least 49 errors. The verb conjugation prefixes with *schwa* are t,j,l,n,m. With the exception of /t/ the prefix has high sonority and should, in most cases, cause a syllable break. Thus the second rule is often subsumed by the third. (This explains the low number of errors when rules 1-2 are ignored.) Since the number of remaining errors was small, we were able to manually identify when rules 1-2 were applicable, thus obtaining an error rate of less than 1%.

Qamatz

The diacritic *qamatz* is most often pronounced /a/ (*big qamatz*). The database contained 199 occurrences where *qamatz* is pronounced /o/ (*small qamatz*). We used two heuristics to identify (some of) them: The *qamatz* was followed by a consonant with the diacritic *hataf-qamatz* (which is always pronounced /o/). Thus, the pattern was /oCo/.

The consonant after the *qamatz* had a *schwa* and the following consonant had a *dagesh* (that indicates germination or strong pronunciation). Thus, the pattern was *qamatz* C1 *schwa* C2 *dagesh*. Hebrew grammar dictates that the *dagesh* is a light *dagesh* and C2 is either ב,ג,ד,כ,פ,ת.

This allowed us to identify 74 cases of /o/ (37.2%). The *small qamatz* is relatively rare, appearing in only 0.6% of all words of the database and in only 3% of the errors.

Conclusions

We have constructed an algorithm to transcribe dotted Hebrew texts to IPA conforming to the observed Israeli Hebrew pronunciation. The algorithm was implemented as a Python 3 program and is available from the author. The program was tested on a large database and the error rate was 11.2%.

We used the database to test how well sonority theory explains the pronunciation of *schwa*, and have formulated a simple alternative algorithm that outperforms the sonority theory algorithm.

References

Bolotzky, S. 2007. The sonority in the phonology of Israeli Hebrew. In *Hebrew and her sisters*, Efrat, M. (ed.). Haifa University Press, 239-248.

Burquest, D. A., and Payne, D. L. 1998. *Phonological analysis: A functional approach*. Dallas, TX: Summer Institute of Linguistics.

O'Grady, W. D., and Archibald, J. 2013. *Contemporary linguistic analysis: An introduction*. (7th ed.). Toronto: Pearson Longman, 70.

Rosen, H. 1957. *Ha-Ivrit Shelanu*, (*Our Hebrew*). Tel Aviv: Am Oved.

	Sonority theory w/o silibants	The alternative algorithm		
		Rules 3-6 w/o silibants	Rules 3-6 with silibants	Rules 1-6 with silibants
Sample size	7449	7449	8612	8612
# errors	871	125	126	77
% error	11.69%	1.68%	1.46%	0.89%

Table 4: Sonority theory and the alternative algorithm for words with ***schwa***