

# Hebrew Word Segmentation using Discourse Data

David Gabay, Ziv Ben-Eliahu and  
Michael Elhadad

# Examples

- מהומות פרצו לאחר הדחת הנשיא בקייב
- המפלגה הציעה לנו את התפקיד

# Examples

- מהומות פרצו לאחר הדחת הנשיא בקייב
- המפלגה הציעה לנו את התפקיד



# Context

Contexts used by existing tools for Hebrew  
Part of Speech tagging and segmentation:

- Immediate – adjacent words
- Local – Sentence.
- Global – Corpus.

# Context

Other contexts out there:

- Immediate – adjacent words
- Local – Sentence.
- Paragraph\Section...
- Document\Article...
- Category\Period...
- ...
- Global – Corpus.

Sentences rarely appear outside of some intermediate context

# Uses of Intermediate Contexts in other Disambiguation Tasks

- English word sense disambiguation
- Chinese word boundary detection
- Chinese and Japanese POS tagging
- ...But not English POS tagging

# Word Segmentation in Hebrew

- Finding prefixes\suffixes
- Major source for ambiguities
  - 50% of tokens have potentially more than one segmentation
- Unknowns and unknown-unknowns
  - “*These are things we do not know we don't know.*” D. Rumsfeld
- Can a sentence-based POS tagger and a good lexicon make do?
  - Unknowns keep on coming: משחן, בקדייטינג, וזובלה
  - Local context may not suffice: ישבתי ואכלתי בצל
  - Distribution of errors
  - ‘Environmental damage’ of errors in segmentation

# “One Segmentation per Document”

- In a typical short text, such as a news article, almost all word types will have the same segmentation throughout
- In a corpus of news articles from Haaretz, out of ~10,000 word-types that re-occur in the same document, only 59 have different segmentations in the same article.

# Determine **Unknowns** using **Context Knowledge**



# D.U.C.K

- <http://sourceforge.net/projects/duck/>
- A general framework
- Input:
  - A list of possible prefixes
  - A hierarchy of contexts
- Output:

For each word type in each document, the probability for each possible segmentation:

*word-type: בקיב document: 1001*

*segmentation probability:*

*[0.3 בקיב 0.7 בקיב]*

# D.U.C.K (2)

Basic method:

- For each ambiguous word type  $v$  in a document, and for each possible segmentation  $v=pw$ , count all words  $u=qw$ , where  $p,q$  are prefixes.
- Assign:

$$\Pr(w) = \frac{\sum_q \text{count}(qw)}{\sum_{v=pw'} \sum_q \text{count}(qw')}$$

- If there is not enough data to determine  $v$ , go up to a higher context

# D.U.C.K (3)

- Not all witnesses are equal: keep weights for each prefix
  - Initialize all weights to 0.5
  - After determining the segmentation of an ambiguous word, update its' prefix weights, if confident enough

# Results

- DUCK was tested primarily on the Wikipedia segmentation corpus
- <http://www.cs.bgu.ac.il/~nlpproj/wiki-seg-corpus/>
- A corpus of Wikipedia articles, partially tagged for segmentation
- Correct segmentation extracted from wiki-links

# Results (2)

- On Wikipedia corpus, ambiguous words only: 76% correct segmentation using article-level context only
- Same, with categories: 85%
- Same, on all words (estimation) 92%
- On Haaretz Corpus: 81%

# Integration with Local Context Tools

- D.U.C.K Probabilities as initial probabilities
  - Only for unseen words
  - For all words
  - For all words, provided there's enough evidence
- Work in parallel with a local model (e.g. HMM)
  - Let the more confident system decide



תודה