בס"ד

Experiments with Language Models for Word Completion and Prediction in Hebrew

Yaakov HaCohen-Kerner,

Asaf Applebaum, Jacob Bitterman

Department of Computer Science

Jerusalem College of Technology – Lev Academic Center

Jerusalem, Israel

Word Completion and Prediction

- Word completion is the task of predicting and automatically completing words that the user is in the process of typing their beginning.

- Word prediction is the task of suggesting words that are likely to follow a given fragment of text.

Word Completion and Prediction (Cont.)

Word completion and prediction
are used in systems
that help people with physical disabilities, search engines, short messages services & mobile phones

- to increase the typing speed
- to decrease the number of keystrokes
- and to reduce writing errors

Evaluation measure for word completion and prediction

The main evaluation measure for word completion & prediction is keystroke saving (KS).

$$KS = (chars - keystrokes)/chars \times 100$$

chars represents the number of characters in the text, including spaces and newlines

keystrokes is the minimum number of key presses required to enter the text using word completion & prediction, including the keystroke to select a prediction from a list and a key press at the end of each utterance.

Our research domain – Word Completion & Prediction for Hebrew

- This research aims to describe various language models (LMs) and combinations created to support word prediction and completion in Hebrew
- This domain has been studied relatively little
- Word prediction for Hebrew is assumed to be more difficult than for English because Hebrew is much richer in its morphology forms
  - The Hebrew language has 70M valid (inflected) forms while English has only 1M [Choueka et al, 2000]
  - In Hebrew, there are up to 7000 declensions for only one stem, while in English there are only a few declensions

The First Hebrew Word Prediction System (1)

Netzer, Adler, & Elhadad (2008) are probably the first to present results of word prediction for Hebrew. They developed a NLP-based system for Augmentative and Alternative Communication (AAC)

They used 3 general kinds of methods:

(1) Statistical methods based on word frequencies and repetition of previous words in the text

(2) Syntactic knowledge: part of speech tags (e.g. nouns, adjectives, verbs, and adverbs) and phrase structures.

(3) Semantic knowledge: assigning categories to words and finding a set of rules that constrain the possible candidates for the next word.

The First Hebrew Word Prediction System (2)

- Netzer et al. used 3 corpuses of varying length (1M words, 10M words & 27M words) to train their system. The best results have been achieved while training a language model (a hidden Markov model) on the 27M corpus.
- Contrary to what they expected, the use of morpho-syntactic information such as part of speech tags didn't improve the results. Furthermore, it decreases the prediction results.
- The best results were obtained using statistical data on the Hebrew language. They report on keystroke saving up to 29% with nine word proposals, 34% for seven proposals, and 54% for a single proposal.

The Second Hebrew Word Prediction System (1)

HaCohen-Kerner and Greenfield (2012) present another Hebrew word prediction system containing the following six components:

(1) Sorted lists of words, frequent nouns, and frequent verbs

(2) 6 corpuses containing around 177M words

(3) 3 LMs (trigram, bigram, and unigram) that were generated using the Microsoft Research Scalable Language-Model-Building Tool and the aforementioned corpora

(4) Results of queries sent to the Google Search Engine

(5) A morphological analyzer generated by MILA

(6) A cache containing the 20 most recently typed words

Short summary of the two previous Hebrew systems

- The KS rates (between 54% to 72%) reported in HaCohen-Kerner and Greenfield (2012) are higher than those (saving of 29% with 9 word proposals, 34% for 7 proposals, and 54% for a single proposal) reported in Netzer et al. (2008).

- However, these 2 systems were trained and tested on different corpora. In any event, it seems that the larger corpora (like in the 2nd system) the higher is the improvement in the prediction results.

Language Models (LMs)

- Language models are applied in various NLP applications, e.g. classification, clustering, IR, MT, and word completion & prediction.
- The most commonly used LMs are the statistical N-gram LMs.
- An n-gram is a contiguous sequence of n items (e.g., letters, words, phonemes, and syllables) from a given sequence of text or speech.
- These LMs try to capture the syntactic and semantic properties of a language by estimating the probability of an item in a sentence given the preceding n-1 items.
- N-gram language models with n=1, 2, 3, & 4 are called unigram, bigram, trigram & fourgram (quadgram) language models, respectively.
- Wandmacher and Antoine (2007), and Trnka and McCoy (2008) show that **n-gram models for word prediction are domain-sensitive**.

Language Models (2)
Previous combinations of LMs

- McMahon (1994) in his Ph.D. dissertation supplies an overview of word based LMs in general and combinations of LMs in particular.
- Beyerlein (1998) has found that an integration of bigram, trigram and quadgram LMs (with 2 additional acoustic models) leads to better results than the 'best' combination of a specific LM and a particular acoustic model.
- Kirchhoff et al. (2006) apply various combinations of LMs for large-vocabulary conversational Arabic speech recognition. They report that combinations of LMs usually have a more significant effect.
- Kimelfeld et al. (2007) use combinations of LMs for XML retrieval. They show that combined LMs generally yield better results in identifying large collections of relevant elements.

The current research – LMs for Word Completion and Prediction in Hebrew

- In this research, we concentrate on KS using various combinations of LMs, unlike the two previous Hebrew systems that use basic LMs.
- We work on 3 corpuses that are dissimilar and smaller than the corpora examined in the two previous systems.
- Thus, no comparisons were made between our system and these systems.

The current research – Examined LMs

We defined, applied, tested and compared 5 kinds of LMs or combinations of them (each kind will be discussed later):

1) **Basic LMs** (unigram, bigram, trigram and quadgram)
2) **Backoff LMs**
3) **Backoff LMs integrated with tagged LMs**
4) **Interpolated LMs**
5) **Interpolated LMs integrated and tagged LMs**

To build the tagged LMs we used a Hebrew tagger built by Meni Adler. This tagger achieved 93% accuracy for word segmentation and POS tagging when tested on a corpus of 90K tokens.

Two first kinds of the examined LMs

(1) **Basic LMs** – The most elementary LMs: unigrams, bigrams, trigrams, and quadgrams

(2) Backoff LMs – Our implemented backoff LM is based on the exclusive use of the highest n-gram basic LM. If this fails to yield results, we then attempt to use the (n–1)-gram LM, etc.

We applied 3 variants of the backoff LMs: the quadgram backoff model, the trigram backoff model and the bigram backoff model

In the event that several proposals have the same highest result, one of the proposals is selected using a random function

(3) Backoff integrated LMs and Tagged LMs

The first variant of this kind of LM uses the backoff LM mentioned in the previous slide. This variant is called conservative since only in a case that there are at least two proposals with the same highest result proposed by the n-gram LM we attempt to choose between them based on the compatible tagged n-gram LM.
If no selection was made, we attempt the (n–1)-gram LM and so on.

The second variant of this kind of LM is Backoff Integrated Tagged LMs with Basic LMs. In contrast to the previous model, this model first activates the tagged n-gram LM. According to the likely POS-tag, it retrieves in context the word that is most likely to fit this POS-tag using the compatible n-gram LM.

(4) Interpolated LMs

This kind of LM is considered to be a general LM as it synthesizes all 4 basic types of n-gram LMs. We defined 4 specific variants:

(1) **Fixed Equal Weights: 0.25** for each n-gram LM

(2) **Fixed Unequal Weights : 0.4, 0.3, 0.2, & 0.1** for the quadgram, trigram, bigram and unigram LMs, respectively.

The reason for giving higher weights to higher n-gram LMs is that they contain larger context environments and therefore are supposed to be more successful.

(3) **Relative weights**: Each n-gram LM gets a weight according to its relative rate of successful predictions and completions of words.

(4) **Like** (3) **with a statistical treatment for the first word in each sentence** based on the distribution of the first word in all sentences.

(5) Interpolated & integrated LMs with tagged LMs

This kind of LM includes 3 variants, which are correspond to variants 2-4 of the previous kind. **However**, this LM also takes into consideration the tagged LMs in a similar way to that presented in the third kind of LM (Backoff Integrated LMs and Tagged LMs).

(1) **Fixed Unequal Weights : 0.4, 0.3, 0.2, & 0.1** for the quadgram, trigram, bigram and unigram LMs, respectively.

The reason for giving higher weights to higher n-gram LMs is that they contain larger context environments and therefore are supposed to be more successful than lower n-gram LMs.

(2) **Relative weights**: Each n-gram LM gets a weight according to its relative rate of successful predictions and completions of words.

(3) **Like** (2) **with a statistical treatment for the first word in each sentence**.

Examined Corpora – 3 different Israeli news web corpuses

Name of Newspaper Corpus	# of word tokens	# of files	General description of the corpus
NRG	551,518	2,500	The online edition of the Israeli newspaper Maariv
TheMarker (TM)	698,577	835	An economic news website that offers ongoing coverage of the capital markets in Israel and globally
A7	880,382	8,724	An Israeli national religious media network that includes

Experimental set

- We have tested 16 specific LMs belonging to 5 kinds of LMs
- Each LM was tested on the 3 mentioned corpora
- We simulated a process of user interaction in the following manner: we went over each word in each sentence in the test corpus. Firstly, we attempted to predict the next word in its entirety. If we failed to do so, we tried to predict a single character at a time until a space or a dot (or any other punctuation mark) was reached or until the next word was correctly proposed.
- The KS results are reported when only one suggestion (with the highest result) is proposed.
- For each specific model we present (in the paper) a unique table

KS results (in %) for the Basic LMs (kind #1)

LM	Cor-pus	Pred-iction	Compl-etion after 1 letter	Compl-etion after 2 letters	Compl-etion after 3 letters	Compl-etion after 7+ letters
1-Grams	NRG	0.34	5.47	11.69	20.66	**28.65**
1-Grams	TM	1.25	4.54	10.14	18.72	**27.20**
1-Grams	A7	0.81	3.81	10.51	19.58	**27.11**
2-Grams	NRG	13.86	23.62	28.96	31.42	**32.76**
2-Grams	TM	11.22	19.41	26.02	28.82	**30.68**
2-Grams	A7	15.28	27.35	36.47	40.09	**41.64**

For 3-grams & 4-grams the KS results were much lower!

Main drawn conclusions for the Basic LMs (kind #1)

- The n-gram models can be ranked according to their KS results (especially the results from 3 known letters) : 2-gram (the best), 1-gram, 3-gram, & 4-gram.
- The limited success of the 3-gram & 4-gram LMs is probably due to the fact that in many cases the discussed word is not presented at all or it lacks the highest frequency. Furthermore, there are many more potential 3 or 4 gram strings than potential 1-grams or 2-gram; thus, it's less likely to predict or complete the correct word.
- The KS results of the 1-gram model are relatively low for fewer than 3 known letters. This is probably because the 1-gram LM for fewer than 3 known letters lacks the necessary context to successfully predict or complete a word.
- In most cases, for all LMs - the more known letters they had the higher KS results they achieved.
- However, even with at least 5 letters, the KS improvement rates were less than 1% for all corpora.

KS results (in %) for the Backoff LMs (kind #2)

LM	Cor-pus	Pred-iction	Compl-etion after 1 let.	Compl-etion after 2 let.	Compl-etion after 3 let.	Completion after 7+ let.
2gram B	NRG	19.13	34.75	43.20	48.56	53.13
2gram B	TM	11.66	20.56	28.63	35.17	41.59
2gram B	A7	14.96	27.46	37.46	43.61	48.23
3gram B	NRG	19.91	27.82	33.38	39.16	45.41
3gram B	TM	16.13	23.92	31.22	37.54	43.77
3gram B	A7	28.35	38.11	45.21	50.13	54.56
4gram B	NRG	20.74	28.81	34.27	40.01	46.17
4gram B	TM	15.78	22.72	29.62	35.79	42.40
4gram B	A7	32.55	40.67	47.27	52.27	56.47

Main drawn conclusions for the Backoff LMs (kind #2)

- **Completion of words:**
  - **NRG** - the **2-gram backoff LM** was the superior LM
  - **TheMarker** - the **3-gram backoff LM** was the superior LM
  - **A7** - the **4-gram backoff LM** was the superior LM
- **Each corpus** has a **different better context environment**
- **Prediction of words:** the **4-gram backoff LM** was found to be the best for two corpora. This finding suggests that for **word prediction** the **4-gram backoff LM** possesses a **superior context environment**
- **All Backoff LMs are better than all basic n-gram models**. This finding means that the simple combination of LMS using a backoff LM is much better than using only one basic LM

KS results (in %) for the Interpolated LMs (kind #4)

LM	Cor-pus	Pred-iction	Comp. after 1 let.	Comp. after 2 let.	Comp. after 3 let.	Comp. After 7+ let.
Fixed Equal weights	NRG	22.72	30.13	35.97	41.45	**47.19**
	TM	16.02	23.03	30.00	36.27	**43.05**
	A7	31.77	39.62	46.57	**51.85**	**56.52**
Fixed unequal weights	NRG	36.02	44.18	49.26	53.79	**58.28**
	TM	16.97	23.88	30.83	37.15	**43.57**
	A7	31.71	39.75	46.41	51.34	**56.31**
Relative weights	NRG	21.16	28.63	34.12	39.7	**45.77**
	TM	17.18	**24.09**	30.91	36.99	**43.41**
	A7	31.73	39.94	46.59	51.34	**56.28**
Rel. weights & treat. for the 1st word	NRG	**39.82**	**48.38**	**53.33**	**57.66**	**61.83**
	TM	**16.91**	23.98	**31.14**	**37.39**	**43.93**
	A7	**31.88**	**40.02**	**46.67**	51.65	**56.62**

Main drawn conclusions for the Interpolated LMs (kind #4)

- **Almost all the leading KS results were achieved** by the **most complex type of Interpolated LM, the LM with relative weights in which the first word in each sentence was treated**.
- The results of **the best Interpolated LM** are also **slightly higher** than those of **the best integrated LM (kind #3)** and **far superior** to those of **the Backoff LMs (kind #2) and the Basic LMs (kind #1)**. The main reason for this might be that **a real synthesis of all 4 basic LMs leads to better results than all other LMs**, **especially in comparison to using only one LM according to the Backoff LMs or a unique Basic LM**

Summary & Conclusions

- We have implemented 5 kinds of LMs (including 16 variants) to support word prediction and completion in Hebrew.
- The best KS results were achieved by the two most complex variants of the most complex kind of LM, the Interpolated and Integrated LMs and Tagged LMs (kind #5).
- Sharing all the strengths in the form of a real synthesis of all 4 basic LMs and the tagged LMs leads to the best results.

Summary & Conclusions (2)

- The improvements of the KS rates for completion of a word after having at least 5 letters was less than 1% for all corpora. That is to say, the contribution of the various LMs and the combinations of LMs is primarily expressed for either prediction or completion of words for less than 5 letters.
- The KS rates for the *TheMarker* corpus (the economic corpus) are significantly lower than those achieved for the other corpora. The reason for this may be that this corpus has relatively larger diversity and contains fewer repetitions of the same n-grams.

Future Work

- Define and apply other combinations of LMs for this task as well as for other domains, applications and languages.
- Examples of possible new LMs are LMs acquired by sampling of n-grams from documents or sampling of documents in order to speed up the construction of LMs.
- Integration of LMs and combinations of LMS with other software components (e.g., queries to search engine, and cache model) in order to improve word prediction and completion in general and for Hebrew in particular.

Thank you very much