CHARACTER WORD EMBEDDING AND POS TAGGING FOR INDIAN LANGUAGES

Anirban Majumdar
Amit Kumar

October 15, 2015

Indian Institute of Technology Kanpur

MOTIVATION

MOTIVATION

- Distributed word representations are proven to be a powerful tool.
- Word embeddings captures syntactic and semantic information about word.
- In task like POS Tagging intra-word information could be very useful which is ignored in word embeddings.
- Character embeddings can be use to capture the intra-word information [1].
- Why not enhance the word embedding to use intra-word information by using character embedding.

RELATED WORK

- Learning Character-level Representations by Santos et al.
- Some results on english language

INCONSIDERABLE	83-YEAR-OLD	SHEEP-LIKE	DOMESTICALLY	UNSTEADINESS	0.0055
INCONCEIVABLE	43-YEAR-OLD	ROCKET-LIKE	FINANCIALLY	UNEASINESS	0.0085
INDISTINGUISHABLE	63-YEAR-OLD	FERN-LIKE	ESSENTIALLY	UNHAPPINESS	0.0075
INNUMERABLE	73-YEAR-OLD	SLIVER-LIKE	GENERALLY	UNPLEASANTNESS	0.0015
INCOMPATIBLE	49-YEAR-OLD	BUSINESS-LIKE	IRONICALLY	BUSINESS	0.0040
INCOMPREHENSIBLE	53-YEAR-OLD	WAR-LIKE	SPECIALLY	UNWILLINGNESS	0.025

INCONSIDERABLE	00-YEAR-OLD	SHEEP-LIKE	DOMESTICALLY	UNSTEADINESS	0.0000
INSIGNIFICANT	SEVENTEEN-YEAR-OLD	BURROWER	WORLDWIDE	PARESTHESIA	0.00000
INORDINATE	SIXTEEN-YEAR-OLD	CRUSTACEAN-LIKE	000,000,000	HYPERSALIVATION	0.000
ASSUREDLY	FOURTEEN-YEAR-OLD	TROLL-LIKE	00,000,000	DROWSINESS	0.000000
UNDESERVED	NINETEEN-YEAR-OLD	SCORPION-LIKE	SALES	DIPLOPIA	±
SCRUPLE	FIFTEEN-YEAR-OLD	UROHIDROSIS	RETAILS	BREATHLESSNESS	-0.00

GOAL

GOAL

- Learning intra-word feature extraction of words using character embedding.
- Enhancing word embedding using the character embedding of the word.
- Using enhanced word embedding to perform task like POS Tagging.

CHALLENGES

CHALLENGES

- Character embedding relatively new field.
- Extracting the morphological information from character embedding
- Use of Enhanced word vectors for NLP tasks such as POS tagging in Indian Languages like Hindi, Bengali

ROADMAP

DATA SET

- Wikipedia english corpus (16 million words, Vocab Size: 70k)
- Training data for POS tagger : wikipedia hindi corpus (200 MB)
- Wikipedia Corpus for Bengali (100 MB)

DATA COLLECTION

- Cleaning english and hindi wikipedia corpus
- Collecting dataset for hindi
- Wiki Extractor for cleaning up the corpus
  `github.com/bwbaugh/wikipedia-extractor`

CHARACTER EMBEDDING RESULT

```
b   b   5.280712    0.173018    14.003992   -2.254795   8.873761    4
1.125260    2.833329    2.280506    -0.351474   2.830920    0.508358
1.437051    -4.516340   -5.023319   1.847007    -4.858042   1.371398
4.526853    -9.200667   1.959274    -0.767200   -0.399730   1.248913
1.747766    8.385695    4.733745    -1.124201   -4.552538   0.922268
b   e   1.747617    -3.959455   0.029893    -1.880294   0.278065    3
4.063965    1.220432    -0.957380   2.115646    1.496750    4.231344
2.864438    -2.265238   -1.497057   -4.475323   3.776992    -0.509295
2.960354    -1.210697   -2.836562   -4.261975   0.722677    -7.269504
0.227397    -0.351846   -2.064200   0.921335    -2.885363   1.559509
b   m   1.529446    -5.822112   -7.563061   -1.504245   6.127498    3
0.005890    1.574102    -3.651192   -0.447519   2.449492    -3.292535
1.335587    -4.275381   -1.900124   0.501332    2.970530    -2.887714
2.541458    6.705114    -1.581140   0.324141    3.367849    2.493709
2.653444    2.598590    1.746207    2.805971    -2.623202   0.036795
```

Figure: Position based character embeddings

USING CWE FOR NLP TASKS : POS TAGGING

- Character Embedding captures the syntactic features
- Can improve the result of tasks like POS tagging and NER
- But how to join the char-level embedding with the word-level one ??

USING CWE FOR NLP TASKS : POS TAGGING

- Options :
  - Average addition to the word embeddings
  - Using CNN approach to get a char-level embedding for a word from the characters of that word
  - More on we can use syllables or affixes instead of character to get the joint embedding

ENHANCED WORD EMBEDDINGS

- Enhancing Word embedding to use intra-word information
- Word embedding from composition of character embeddings
  - Average Addition [2] character embedding vector without feature extraction

$$\mathbf{x}_j = \frac{1}{2}(\mathbf{w}_j + \frac{1}{N_j}\sum_{k=1}^{N_j}\mathbf{c}_k).$$

  - Feature Extraction using CNN and adding information to word embeddings
- Using the joint learned embedding for the purpose like POS tagging

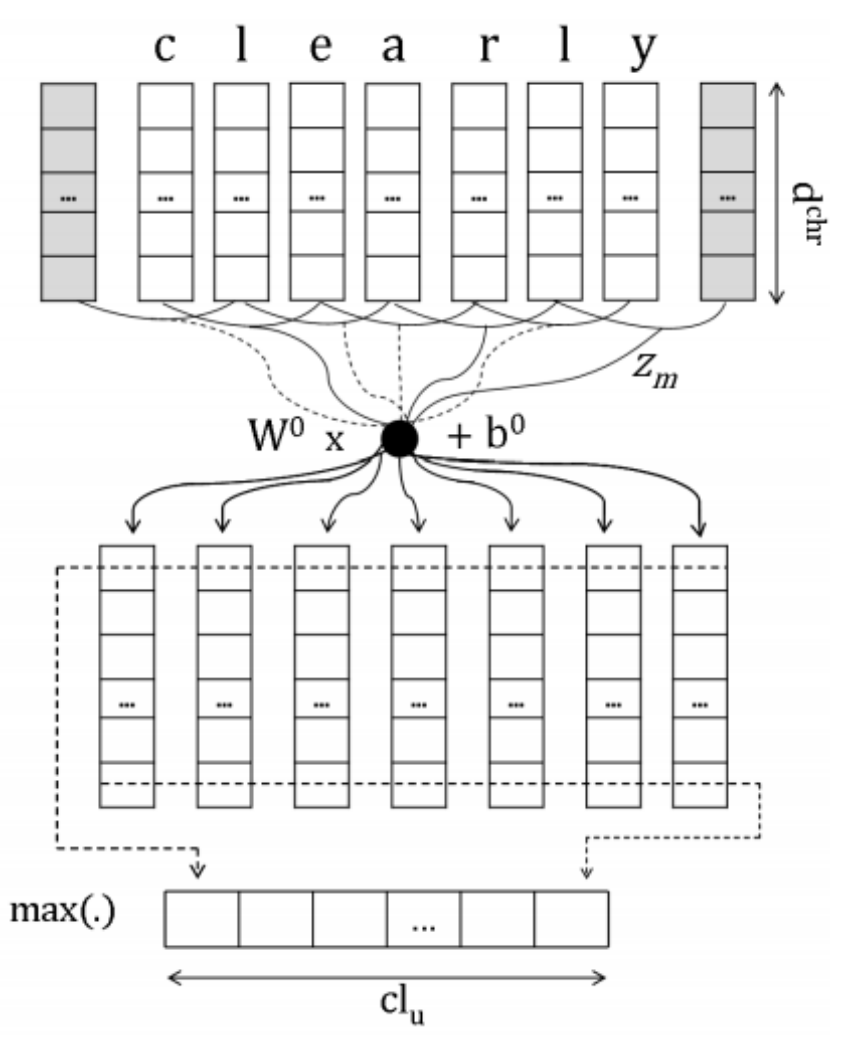
SOME RESULTS ON AVERAGE ADDITON

Table 3: Evaluation accuracies (%) on analogical reasoning.

Method	Total	Capital	State	Family
CBOW	54.85	51.40	66.29	62.92
+CWE	58.24	53.32	66.29	70.00
+CWE+P	60.07	54.36	66.29	73.75
Skip-Gram	69.14	62.78	82.29	80.83
+CWE	68.04	63.66	81.14	78.75
+CWE+P	72.07	65.44	**84.00**	**84.58**
GloVe	67.44	69.22	58.05	69.25
+CWE	70.42	70.01	64.00	76.25
+CWE+P	**72.99**	**73.26**	65.71	81.25

CHARACTER EMBEDDINGS FEATURE EXTRACTION

- Extracting character embeddings for the given corpus
- Feature extraction from character embeddings using CNN



POS TAGGING FOR HINDI

- Previous work for POS tagging is mostly based on Statistical or Rule Based Model
- Can improve the results using the joint embeeding
- Advantage : Less hand-crafted features

NEAREST NEIGHBOURS FOR CWE EMBEDDING WORDS FOR WIKI

- railways : motorways (20.571344), rail (21.448918), railway (21.594830), trams (21.744342),tramways (21.434643)

- primarily : mainly (11.726825), mostly (12.344781), principally (15.456143), chiefly (15.708947), largely (15.779496), and (16.920006), secondarily (17.022827)

REFERENCES

Cicero D. Santos and Bianca Zadrozny. “Learning Character-level Representations for Part-of-Speech Tagging”. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). Ed. by Tony Jebara and Eric P. Xing. JMLR Workshop and Conference Proceedings, 2014, pp. 1818–1826. URL: http://jmlr.org/proceedings/papers/v32/santos14.pdf.

Zhiyuan Liu Maosong Sun
Huanbo Luan Xinxiong Chen Lei Xu. “Joint Learning of Character and Word Embeddings”. In: (2015).

QUESTIONS?

APPENDIX

CHAR-LEVEL EMBEDDING USING CNN - DETAILS

- Produces local features around each character of the word
- Combines them to get a fixed size character-level embedding
- Given a word w composed of M characters $c_1, c_2, ..., c_M$, each c_M is transformed into a character embedding r_m^{chr} . Them input to the convolution layer is the sequence of character embedding of M characters.

CHAR-LEVEL EMBEDDING USING CNN - DETAILS

- Window of size kchr (character context window) of successive windows in the sequence of $r_1^{chr}, r_2^{chr}, ..., r_M^{chr}$
- The vector z_m (concatenation of character embedding m)for each character embedding is defined as follows :

$$z_m = (r^{chr}_{(m-(k^{chr}-1)/2)}, ..., r^{chr}_{(m+(k^{chr}-1)/2)})T$$

CHAR-LEVEL EMBEDDING USING CNN - DETAILS

- Convolutional layer computer the jth element of the character embedding rwch of the word w as follows:

$$[r^{wch}]_j = max_{1<m<M}[W^0 z_m + b^0]_j$$

- Matrix W^0 is used to extract local features around each character window of the given word
- Global fixed-sized feature vector is obtained using max operator over each character window

CHAR-LEVEL EMBEDDING USING CNN - DETAILS

- Parameter to be learned :
  - $W^{chr}, W^0 and b^0$
- Hyper-parameters :
  - d^{chr} : the size of the character vector
  - cl_u : the size of the convolution unit
    (also the size of the character-level embedding)
  - k^{chr} : the size of the character context window