ARISTOTLE UNIVERSITY OF THESSALONIKI

Figure 1.A.U.Th. logo. Retrieved from https://www.auth.gr/en/logo

Natural Language Processing

Andrei Volodin
January 2020

DIM102

NATURAL LANGUAGE PROCESSING (NLP)

CLOSEST NEIGHBORS

Computer Science (CS)

Speech Processing

Artificial Intelligence (AI)

Machine Learning (ML)

Computational linguistics

NATURAL LANGUAGE PROCESSING

DEFINITION:

“is the set of methods for making human language accessible to computers” (Eisenstein, 2019)

Example tasks

Easy
- Spell Checking • Keyword Search • Finding Synonyms

Medium
- Parsing information from websites, documents, etc.

Hard
- Machine Translation (e.g. Translate Chinese text to English)
- Semantic Analysis (What is the meaning of query statement?)
- Coreference (e.g. What does "he" or "it" refer to given a document?)
- Question Answering (e.g. Answering Jeopardy questions).

source: s224d.stanford.edu

NATURAL LANGUAGE PROCESSING

Unix **wc** program

“is used to count the total number of bytes, words, and lines in a text file.”

“When used to count bytes and lines, wc is an ordinary data processing application.”

However, when it is “used to count the words in a file it requires knowledge about what it means to be a word, and thus becomes a *language processing system*.”

(Jurafsky, 2008)

Word to Vector representation

Word meaning is defined in terms of vectors

- In all subsequent models, including deep learning models, a word is represented as a dense vector

$$\textit{linguistics} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \end{pmatrix}$$

Natural language processing is fun. $= \begin{pmatrix} -0.132 \\ 1.129 \\ 0.827 \\ 0.110 \\ -0.527 \\ 0.156 \\ 0.349 \\ -0.286 \end{pmatrix}$

Figure 5. Word2vec. Retrieved from https://cs224d.stanford.edu/lectures/CS224d-Lecture2.pdf

NATURAL LANGUAGE PROCESSING

Stanford CoreNLP toolkit

“an extensible pipeline that provides core natural language analysis. This toolkit is quite widely used, both in the research NLP community and also among commercial and government users of open source NLP technology”

“a Java (or at least JVM-based) annotation pipeline framework, which provides most of the common core natural language processing (NLP) steps, from tokenization through to coreference resolution” (Manning, 2014).

The toolkit was designed for internal use and was released to opensource in 2010.

(Manning, 2014)

NATURAL LANGUAGE PROCESSING: Stanford CoreNLP toolkit

Example inputs

```
$ cat sentiment.txt
I liked it.
It was a fantastic experience.
The plot move rather slowly.
$ java -cp "*" -Xmx2g edu.stanford.nlp.pipeline.StanfordCoreNLP -annotators
tokenize,ssplit,pos,lemma,parse,sentiment -file sentiment.txt
```

Outputs

```
$ grep sentiment sentiment.txt.xml
<sentence id="1" sentimentValue="3" sentiment="Positive">
<sentence id="2" sentimentValue="4" sentiment="Verypositive">
 <sentence id="3" sentimentValue="1" sentiment="Negative">
```

NATURAL LANGUAGE PROCESSING

Google Cloud: Text-to-Speech API https://cloud.google.com/text-to-speech/

Google uses DeepMind’s Wavenet

Google Assistant: Speech Recognition

And DeepMind’s Wavenet uses Softmax which helps classifying

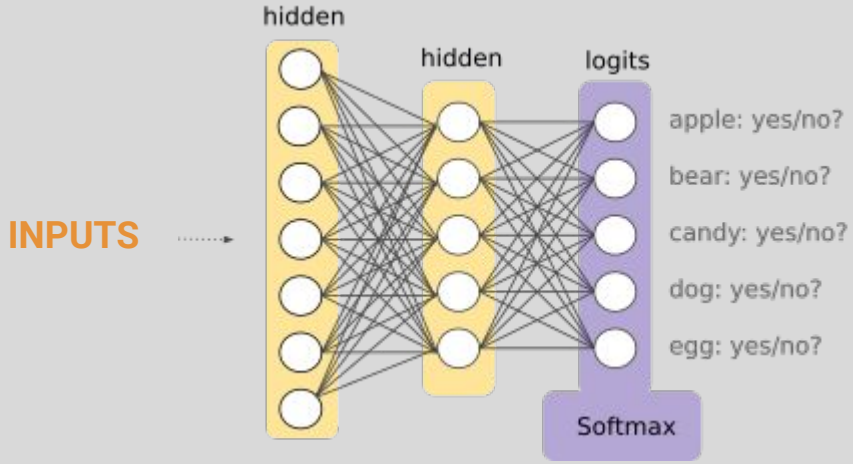

Figure 2. Softmax. Image. Retrieved from https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

NATURAL LANGUAGE PROCESSING

Stanford Question Answering Dataset (SQuAD)

is a reading comprehension dataset, consisting of questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or *span*, from the corresponding reading passage, or the question might be unanswerable

(Rajpurkar, 2019)

In 2016, Rajpurkar released the the Stanford Question Answering Dataset(SQuAD 1.0) which consists of 100K question-answer pairs each with a given context paragraph and it soon becomes a standard test for the reading comprehension task with public leaderboard available. In 2018, the team further released SQuAD 2.0, which contains over 50,000 unanswerable questions that post a much harder requirement on model development

(Zhang)

NATURAL LANGUAGE PROCESSING

Google BERT: **B**idirectional **E**ncoder **R**epresentations from **T**ransformers

Google ALBERT: A Light BERT for Self-Supervised Learning of Language Representations

a new technique for NLP pre-training

“BERT also learns to model relationships between sentences by pre-training on a very simple task that can be generated from any text corpus:”

Sentence A = `The man went to the store.`
Sentence B = `He bought a gallon of milk.`
Label = `IsNextSentence`

Sentence A = `The man went to the store.`
Sentence B = `Penguins are flightless.`
Label = `NotNextSentence`

(Devlin, 2018)

Finding Syntax with Structural Probes

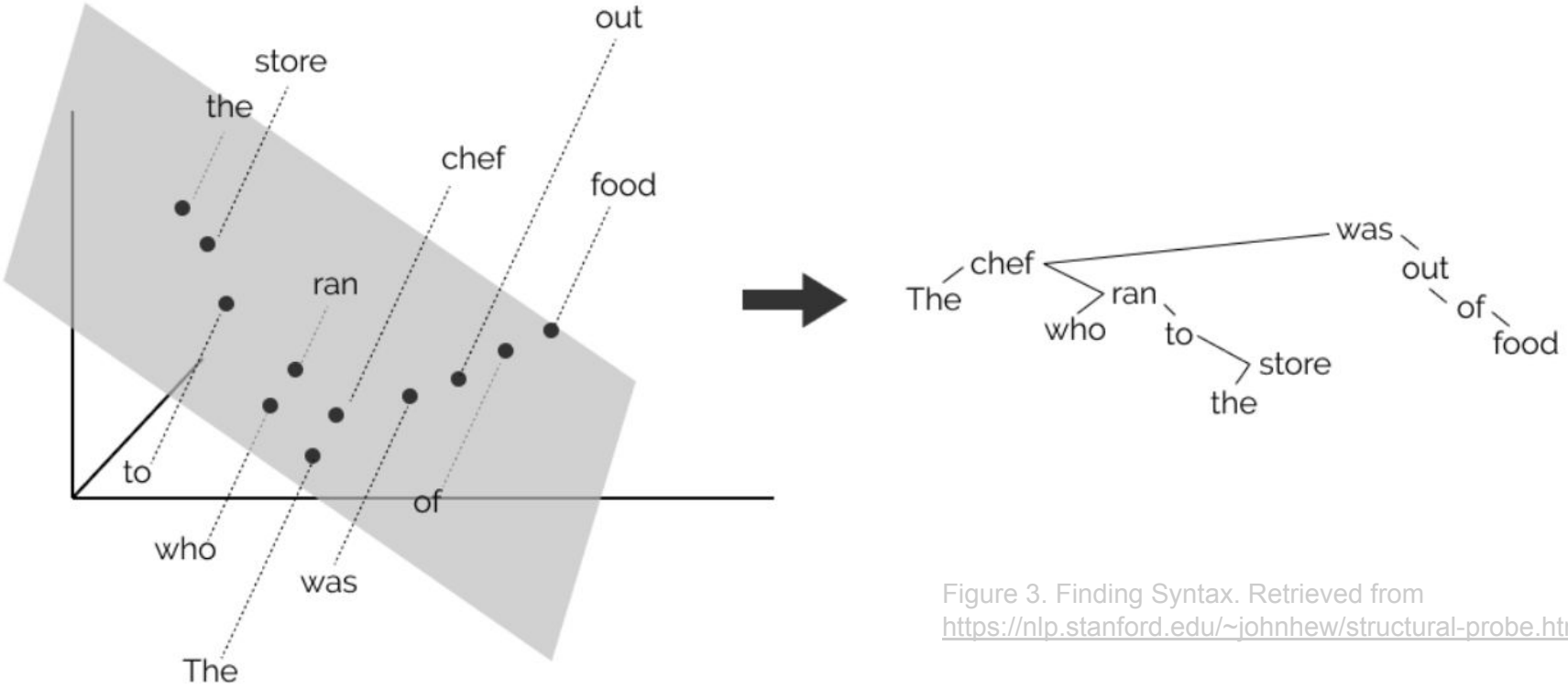

Figure 3. Finding Syntax. Retrieved from https://nlp.stanford.edu/~johnhew/structural-probe.html

Sequence - to - Sequence (Seq2Seq)

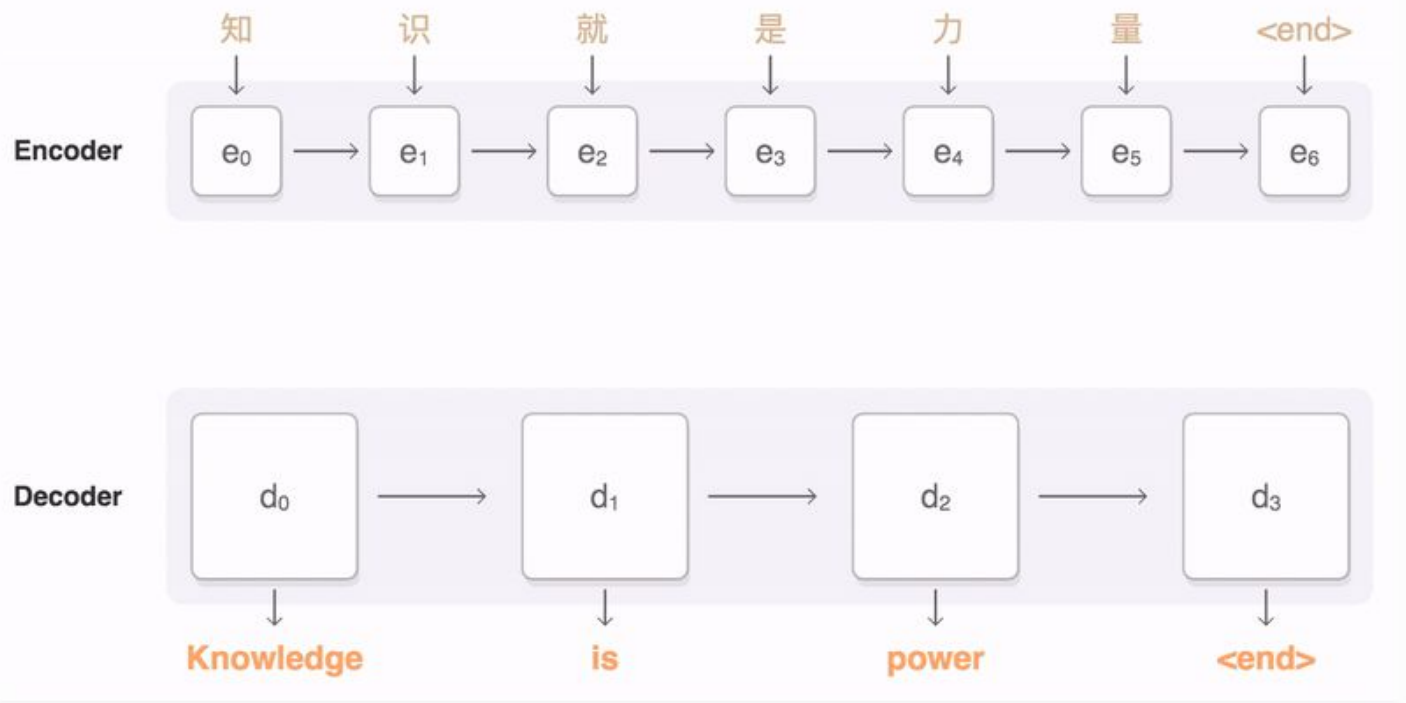

Figure 4. Seq2Seq. Retrieved from https://google.github.io/seq2seq/

REFERENCES

Figure 1.A.U.Th. logo. Retrieved from https://www.auth.gr/en/logo

Manning, C., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations (pp. 55-60).

Figure 2. Softmax. Image. Retrieved from https://developers.google.com/machine-learning/crash-course/multi-class-neural-networks/softmax

Lecture Notes. CS224d. Retrieved from http://cs224d.stanford.edu/lecture_notes/notes1.pdf

Eisenstein, J. (2019). Introduction to natural language processing. Mit Press.

Li, Y., & Zhang, Y. Question Answering on SQuAD 2.0 Dataset.

Daniel Jurafsky and James H. Martin. 2008. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition. Second Edition. Prentice Hall.

Devlin, J., & Chang, M. W. (2018). Open Sourcing BERT: State-of-the-Art Pre-training for Natural Language Processing. Google AI Blog, November, 2.

REFERENCES

Figure 3. Finding Syntax. Retrieved from https://nlp.stanford.edu/~johnhew/structural-probe.html

Hewitt, J., & Manning, C. D. (2019, June). A structural probe for finding syntax in word representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 4129-4138).

Figure 4. Seq2Seq. Retrieved from https://google.github.io/seq2seq/

Figure 5. Word2vec. Retrieved from https://cs224d.stanford.edu/lectures/CS224d-Lecture2.pdf