A Survey of Word Embedding Algorithms for Textual Data Information Extraction

E. Vušak*, V. Kužina* and A. Jović*

* University of Zagreb, Faculty of Electrical Engineering and Computing, Unska 3, 10 000 Zagreb, Croatia

eugen.vusak@fer.hr

As part of:
"Digital platform for ensuring data privacy and prevention of malicious manipulation of the personal data – AIPD2"

MIPRO 2021
Summary

Textual data

Word embedding

- Context-level learned models
- Subword-level learned models
- Character-level learned models
- Contextualized models
Textual data
Textual data and humans
Textual data and humans

Natural language
Textual data and humans

Natural language

Majority of knowledge
Textual data and humans

Natural language

Majority of knowledge

Most complex invention?
Textual data and computers
Textual data
and computers

Not numbers
Textual data and computers

Not numbers

Multiple “layers” of information
Not numbers

Multiple “layers” of information

Word embeddings
Word embeddings
Word embeddings

Dense vector representations of words where similar words have similar embedding vectors
Word embeddings

Dense vector representations of words where similar words have similar embedding vectors
Word embeddings

Dense vector representations of words where similar words have similar embedding vectors
Context-level learned models
Context-level learned models

“You shall know a word by the company it keeps” - John Rupert Firth
“You shall know a word by the company it keeps” - John Rupert Firth

Words are embedded based on their context
“You shall know a word by the company it keeps” - John Rupert Firth

Words are embedded based on their context

Example: "A bee is buzzing around."
--- "A fly is buzzing around"
“You shall know a word by the company it keeps” - John Rupert Firth

Words are embedded based on their context

Example: "A bee is buzzing around." --- "A fly is buzzing around"
Context-level learned models

Notably:

- **NNLM** - Neural Network Language Model
- **SENNA** - Semantic/syntactic Extraction using a Neural Network Architecture
- **Word2Vec (CBOW, Skip-gram)**
- **GloVe** - Global Vectors for Word Representation
Subword-level learned models
Subword-level learned models

Subword information is considered for an embedding of a word
Subword information is considered for an embedding of a word

Able to capture morphological structures
Subword-level learned models

Subword information is considered for an embedding of a word

Able to capture morphological structures

Example: “breakable”, “biased” → “unbreakable”, “unbiased”
Subword-level learned models

Notably:

- **MorphoRNN** - Morphological Recurrent Neural Network
- **BPE** - Byte Pair Encoding
- **FastText**
Character-level learned models
Character-level learned models

Characters are used for word embedding
Character-level learned models

Characters are used for word embedding
Languages with logographic system of characters
Character-level learned models

Characters are used for word embedding
Languages with logographic system of characters
Learn more complex morphological structure
Character-level learned models

Notably:

- **CWE** - Character-enhanced Word Embedding model
- Methods based on **CNNs**
Contextualized models
Contextualized models

Train vs inference

Multiple words used to calculate vector representation
Contextualized models

Train vs inference
Multiple words used to calculate vector representation
Same word - multiple meanings
Contextualized models

Train vs inference
Multiple words used to calculate vector representation
Same word - multiple meanings
Example: “I can see the can.”
Contextualized models

Notably:

● ELMo - Embedding from Language Models
● GPT - Generative Pre-Training
● BERT - Bidirectional Encoder Representations for Transformers
● XLNet
Conclusions

- No silver bullet
Conclusions

- No silver bullet
- Context-level learned models
  - Fast and efficient
  - Unable to handle OOV
Conclusions

● No silver bullet
● Context-level learned models
  ○ Fast and efficient
  ○ Unable to handle OOV
● Subword-level learned models
  ○ Morphological structure
  ○ Better OOV
Conclusions

- No silver bullet
- Context-level learned models
  - Fast and efficient
  - Unable to handle OOV
- Subword-level learned models
  - Morphological structure
  - Better OOV
- Character-level learned models
  - More complex structures
  - Eliminates OOV
Conclusions

- No silver bullet
- Context-level learned models
  - Fast and efficient
  - Unable to handle OOV
- Subword-level learned models
  - Morphological structure
  - Better OOV
- Character-level learned models
  - More complex structures
  - Eliminates OOV
- Contextualized models
  - Different meanings for same words
  - Price to pay
Questions?