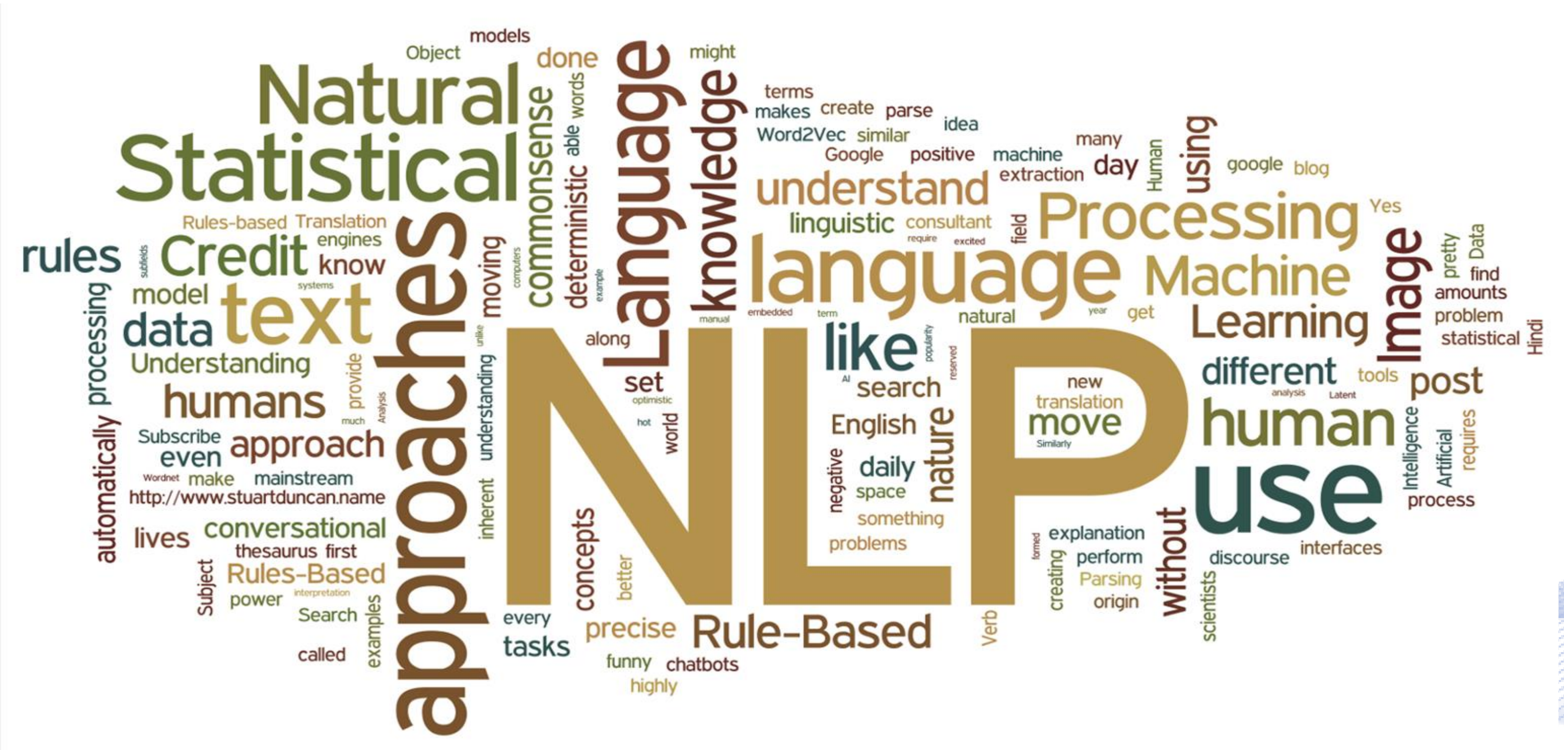
Natural Language Processing for Natural Disasters

P.Giannouris, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
polydoros@ece.auth.gr
www.aiia.csd.auth.gr

Artificial Intelligence & Information Analysis Lab

Introduction



What is NLP ?

Introduction

Natural Language is the way we, humans, communicate with each other.

- Speech and text.

- Given its importance, we must have methods to understand and reason about natural language.

Introduction

Natural Language Processing (***NLP***) is the automatic manipulation (analysis or transformation) of natural language (text and speech).

- It has been around for more than 70 years.
- It grew out of the field of linguistics with the rise of computers.

NLP can translate text

Ελληνικά ⇄ Αγγλικά

Η μηχανική μετάφραση εμφανίστηκε τη δεκαετία του 50

I michanikí metáfrasi emfanístike ti dekaetía tou 50

Machine translation appeared in the 50s

NLP can answer our questions

- What is the weather like today?
- Who is Noam Chomsky?
- How many hours are there in a year?
- Who won the 2022 US elections?

IBM's Watson competed against Jeopardy! champions.

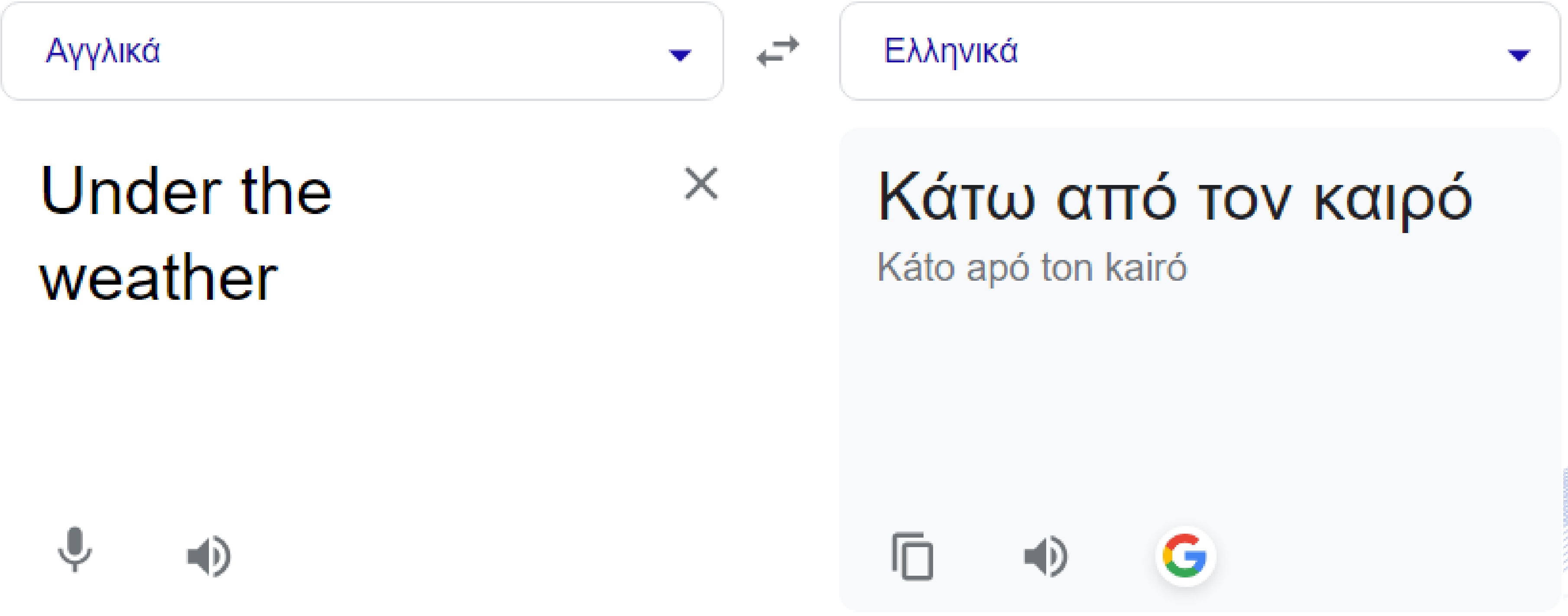
Artificial Intelligence & Information Analysis Lab

NLP can aid in Natural Disasters

Artificial Intelligence &
Information Analysis Lab

Or can it?

NLP can’t translate text



NLP can’t answer our questions

Why is NLP hard?

Natural language is:

- messy
- ambiguous
- changing and evolving
- not always well described by formal rules.
- It has complex structure.
- It can map almost the entire human knowledge.

Thus, it is hard to analyze and transform (e.g., translate) language data.

How does NLP work?

Artificial Intelligence & Information Analysis Lab

Tokenization

vocabulary - all unique words in a source of text
token - an integer value assigned to each word in the vocabulary

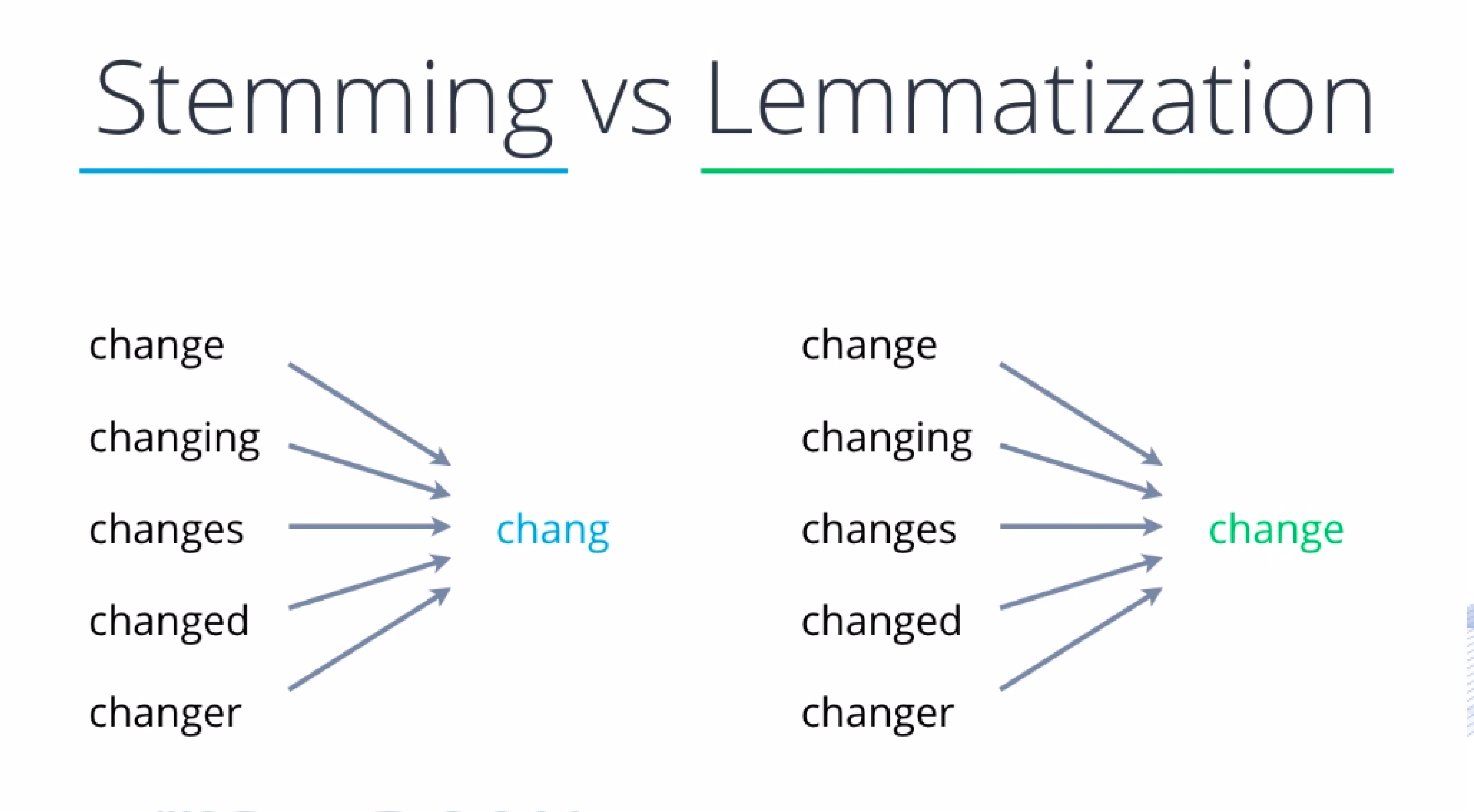
token dictionary

{'the': **0**, 'of': **1**, 'so': **2**, 'then': **3**, 'you': **4**, … 'learn': **3191**, … 'artificial': **30297**… }

sample text | **tokenized text**

"the pettiness of the whole situation" → [0, 121241, 1, 0, 988, 25910]

Preprocessing



Stop Word Removal

[‘~~And~~’, ‘then’, ‘~~the~~’, ‘quick’, ‘brown’, ‘fox’ , ‘jumps’, ‘over’, ‘~~the~~’ , ‘lazy’, ‘dog’] → [‘then’, ‘quick’, ‘brown’, ‘fox’ , ‘jumps’, ‘over’, ‘lazy’, ‘dog’]

Early Approaches to NLP

Rule based Systems

Define hand made linguistic rules

Capture pattens and semantics

Pros	Cons
• Control	• Not scalable
• Transparent and Interpretable	• Ambiguous Language
	• Maintenance

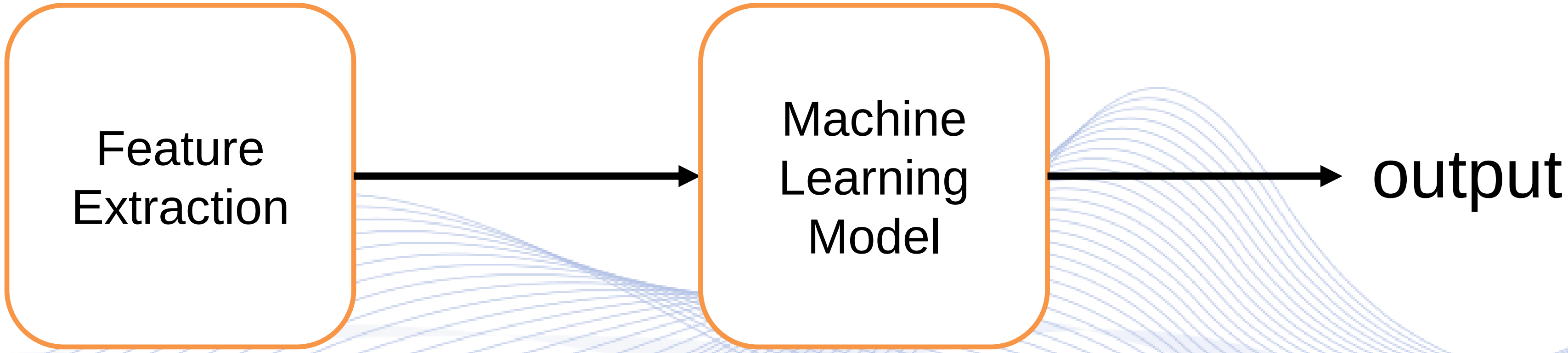
Rule based Systems

pos
I love this lecture → pos
neu
neg

pos
neu
I hate this lecture → neg
neg

pos
I saw this lecture → neu
neu
neg

Machine Learning



Features

- **How many times did each word appear in the sentence?**

  Bag of Words or word counts.

- **What about word pairs (triplets, quartets …)?**

  N-gram features meaning N consecutive words.

- **What makes a word important?**

  TF-IDF: A word is important in a piece of text the more often it appears **while** not appearing often in different texts.

Features

- Term frequency: $TF(t,d) = \frac{number\ of\ time\ t\ appears\ in\ d}{total\ terms\ in\ d}$

- Document Frequency: $\mathrm{IDF}(t) = log\frac{number\ of\ documents}{1+df}$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

Problems

- No concept of **similar** words.

He bought an apple
He purchased an orange

- Features are domain specific

  Useful features in one domain may not provide information in others. Finding new features for every task/domain is costly.

Word representations

Distributional Hypothesis

- Words which are synonyms tend to co-occur in the same environment.

The amount of ***word meaning*** difference between two words corresponds roughly to the difference in the environments.

Instead of identifying each word by a number, find a way to encode its meaning through context.

Word representations

Term-context matrix

- *Use a large corpus to study the use of each word.*
- Each word is identified by its co-occurrences with every other word in the corpus (rows).

- ***Co-occurrence probability***:

$$P(k, l) = \Pr\{\text{vocabulary word } k \text{ co-occurs with word } l\}.$$

Word representations

Term-context matrix

- Similar words are closer in vector space.

	aardvark	...	computer	data	result	pie	sugar	...
cherry	0	...	2	8	9	442	25	...
strawberry	0	...	0	0	1	60	19	...
digital	0	...	1670	1683	85	5	4	...
information	0	...	3325	3982	378	5	13	...

Captures similarity

What's wrong with this approach?

Word embeddings

Word embeddings embed words in a vectorial space.

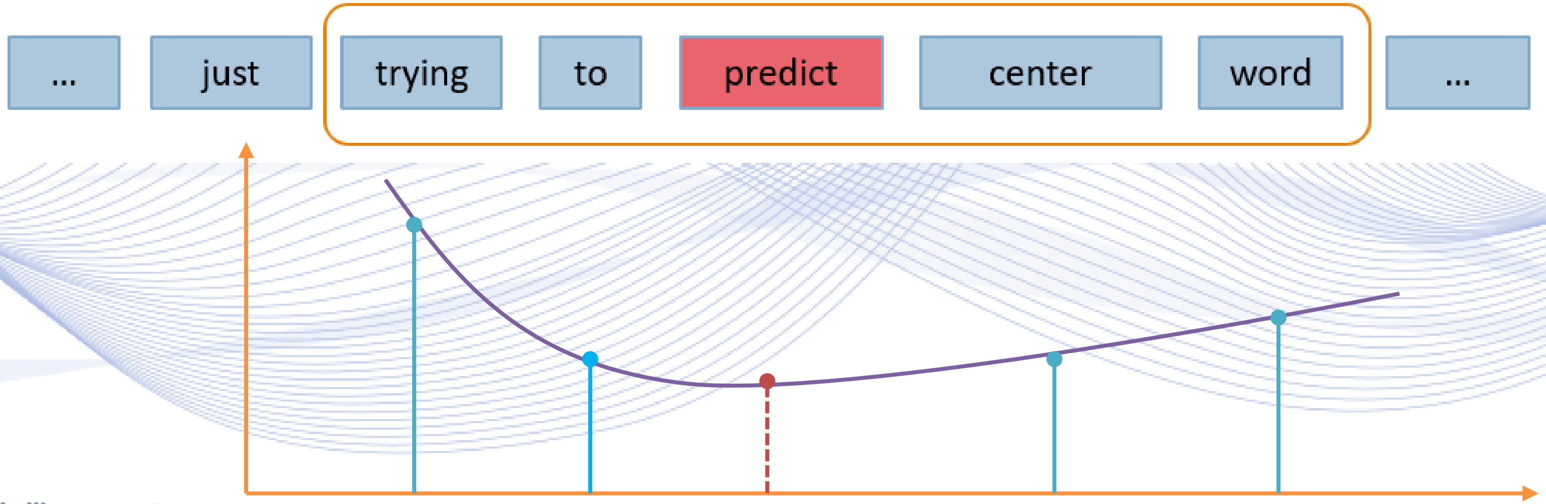
- Fixed length vectors.
- Essentially *dense* word representations.
- Utilize the distributional hypothesis.
- Word embeddings can be learned to satisfy certain optimization criteria.

Word embeddings

Word2Vec

Two-layer NN trained to reconstruct linguistic context of words.

- Training is performed with pairs of context-target words.
- 2 training variations.

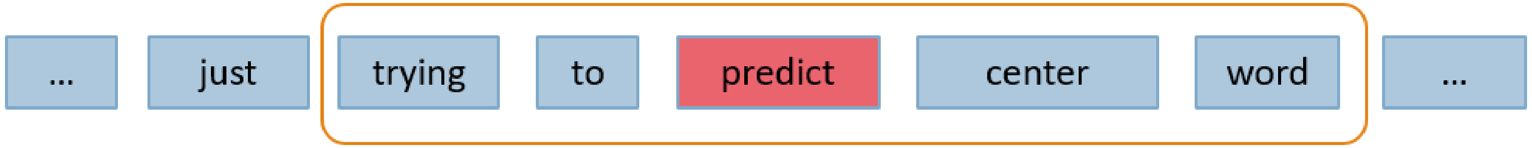


Word embeddings

Continuous Bag-of-Words (***CBOW***):

- Predict center word given surrounding words

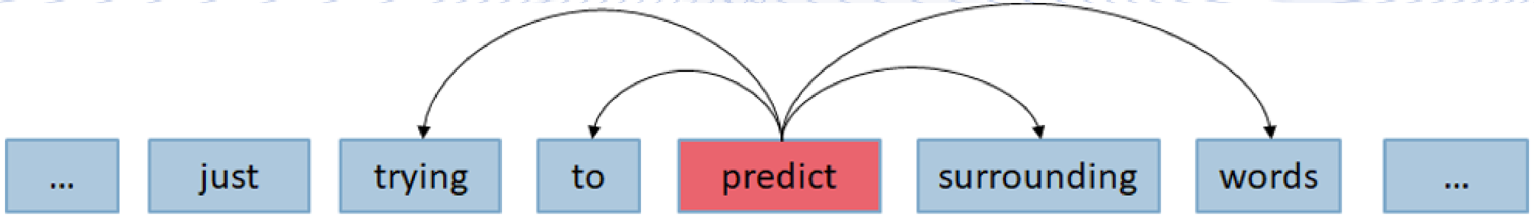
input1	input2	input3	input4	output
trying	to	center	word	predict



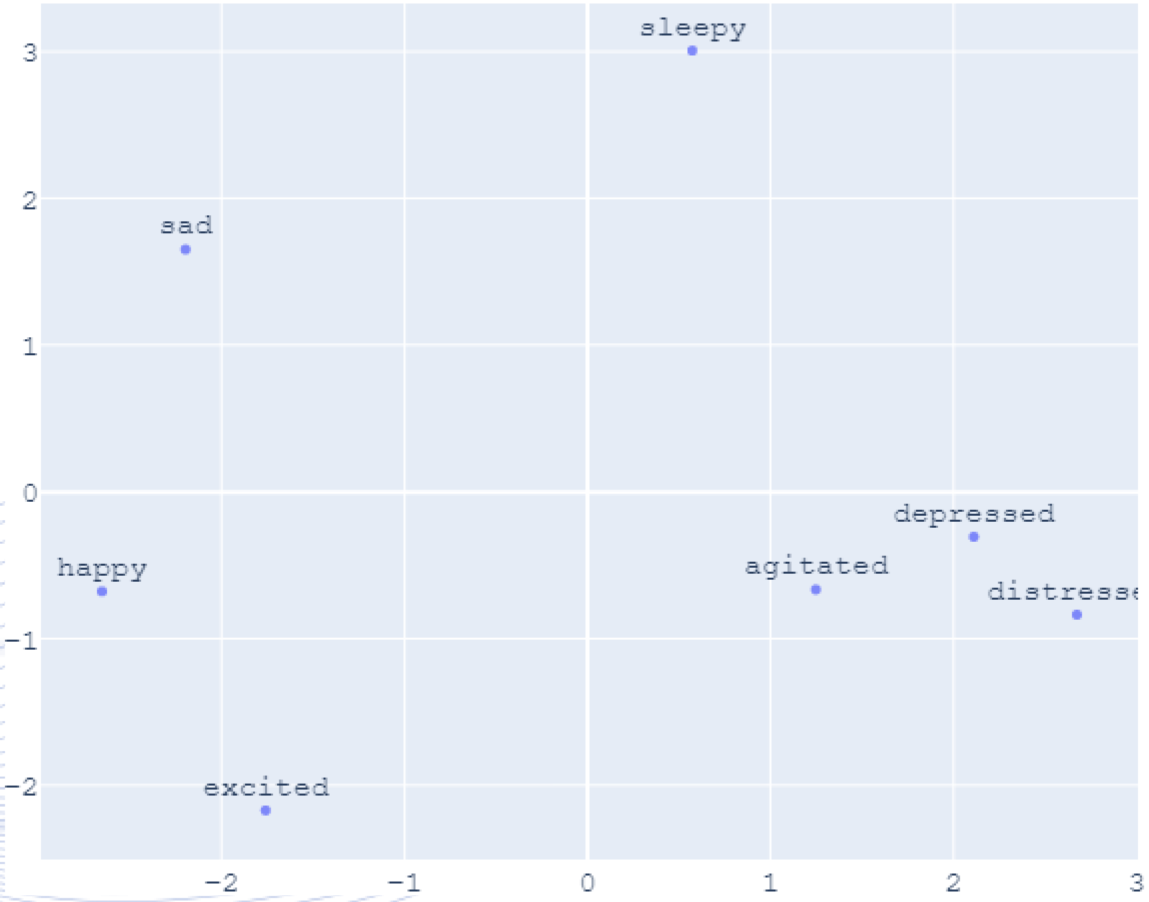
Skip-Gram:

- Predict surrounding words given center word

Input	output1	output2	output3	output4
predict	trying	to	center	word



Word embeddings



Principal components analysis of word embeddings.

Problems solved

- Similar words close in feature space

He bought an apple He purchased an orange	✓

- One word can have multiple meanings

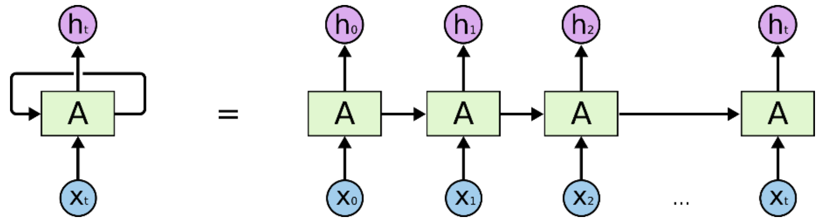
Turn left at the intersection She left after 5 minutes	✗

Artificial Intelligence & Information Analysis Lab

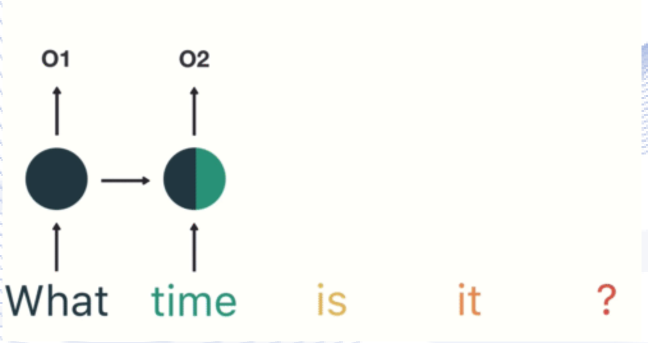
Neural NLP tools

Recurrent Neural Networks

- Good for analyzing ***sequential data*** (like text).
- Text input is given sequentially.
- Each RNN node contains past information in its ***hidden states***.
- Text analysis considers information of previous nodes along with current text input.



Source: colah's blog



Source: link

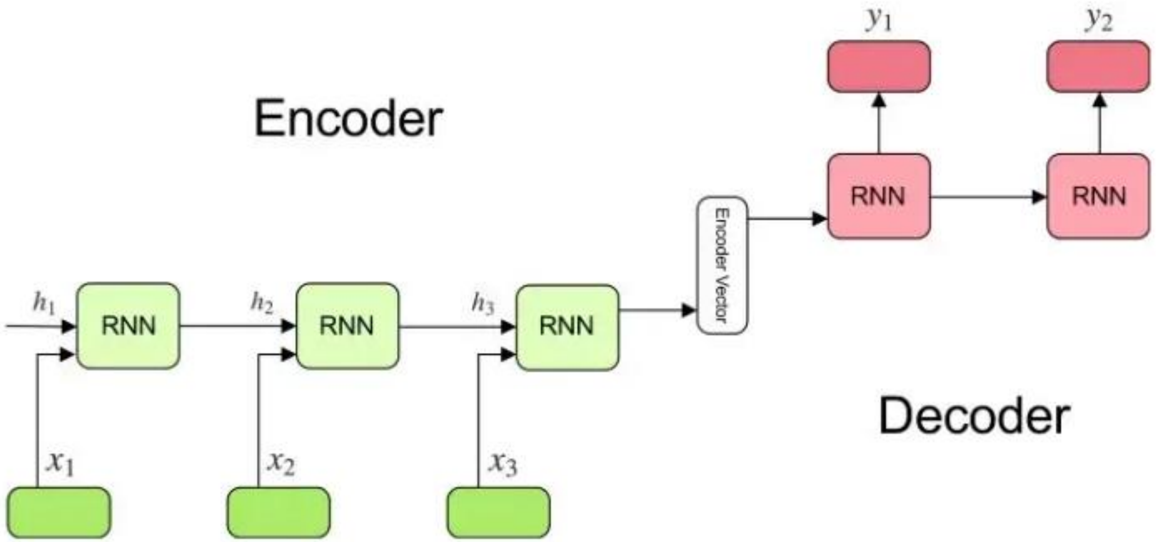
Artificial Intelligence & Information Analysis Lab

Neural NLP tools

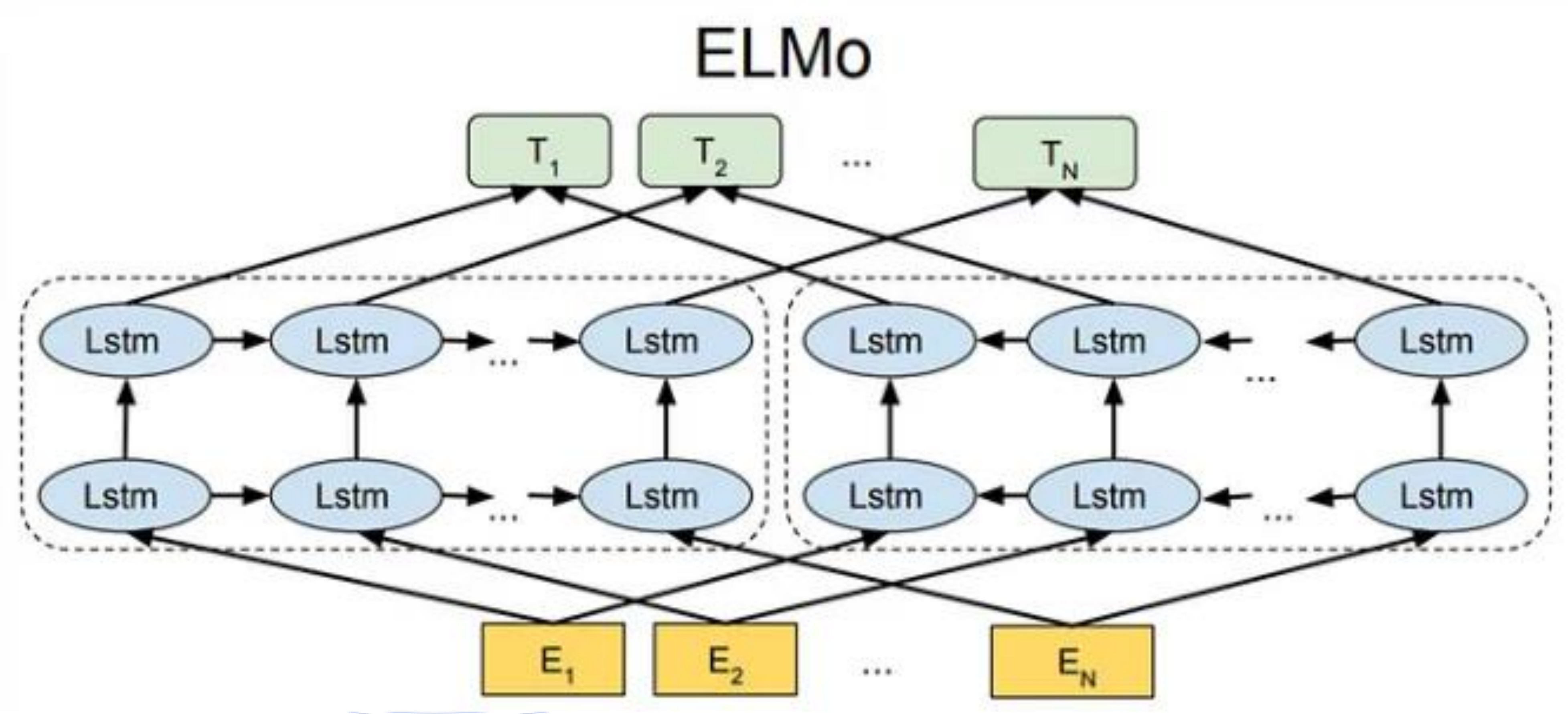
Recurrent Neural Networks

RNNs can also be used to produce new sequences.

- First pass the input sentence through the RNN and ***encode*** its meaning in a vector.
- Then ***decode*** the vector into new sequence.



ELMo


ELMo
T_1 T_2 ... T_N
Lstm Lstm ... Lstm Lstm Lstm ... Lstm
Lstm Lstm ... Lstm Lstm Lstm ... Lstm
E_1 E_2 ... E_N

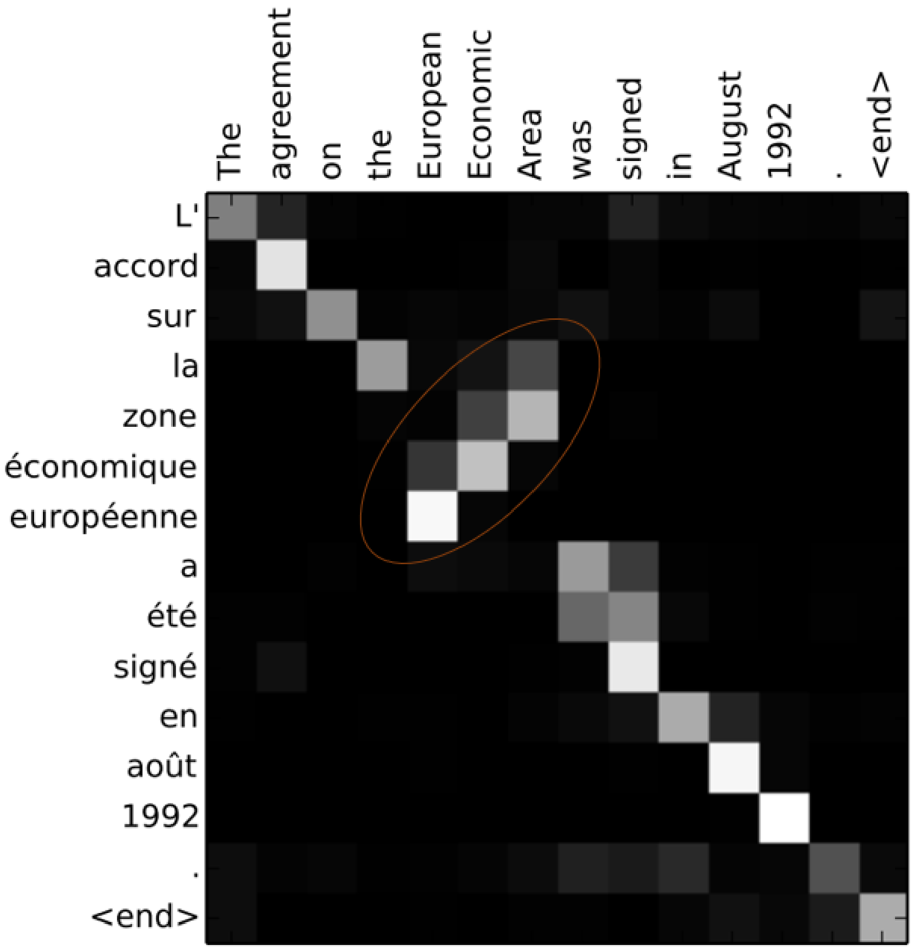
Attention

Attention [BAH2014]

- RNNs “forget” in long sequences.
- RNNs can focus on certain key words using attention:
  - A decoder state (***query***) and encoder states (***keys***).
  - For each query-key pair calculate a weight.
  - Use weighted sum of ***value*** vectors (usually encoder states).

Attention answers the question: *How does the word I’m trying to predict in the output correlate with each word of the input?*

Attention

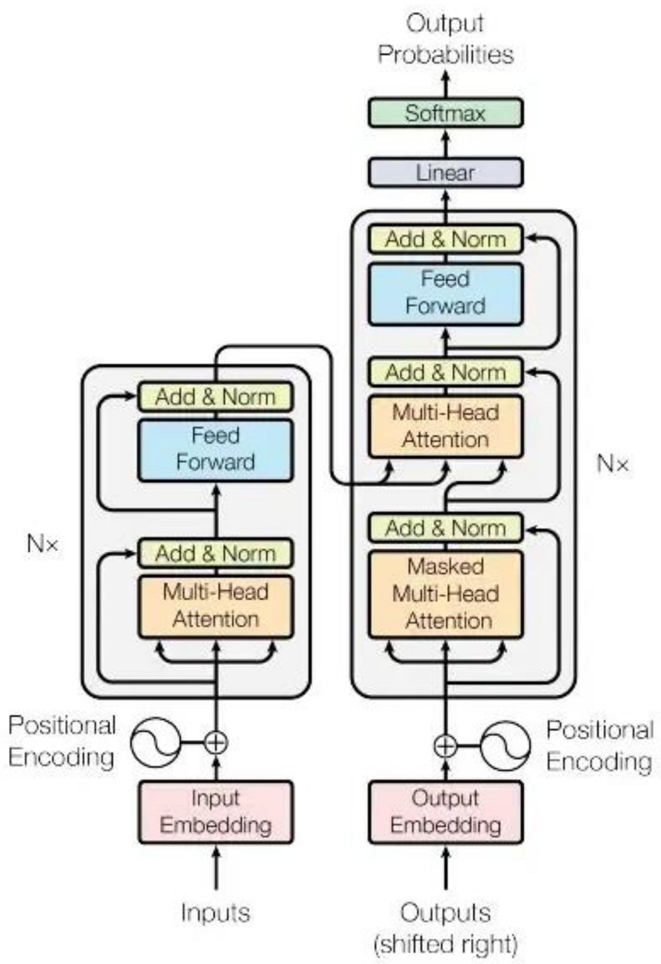

The
agreement
on
the
European
Economic
Area
was
signed
in
August
1992
.
<end>
L'
accord
sur
la
zone
économique
européenne
a
été
signé
en
août
1992
.
<end>

Transformers

Attention is all you need [VAS2017].

Transformers:

- They employ self-attention and cross-attention mechanisms.
- They do not suffer from RNN limitations.
- They are trained in parallel.
- They can spot long dependencies.



Language models

Language models assign a probability distribution over a sequence of words.

$$P(w_i = w \mid w_1, \dots, w_n)$$

Language models **vs** Word embedding models

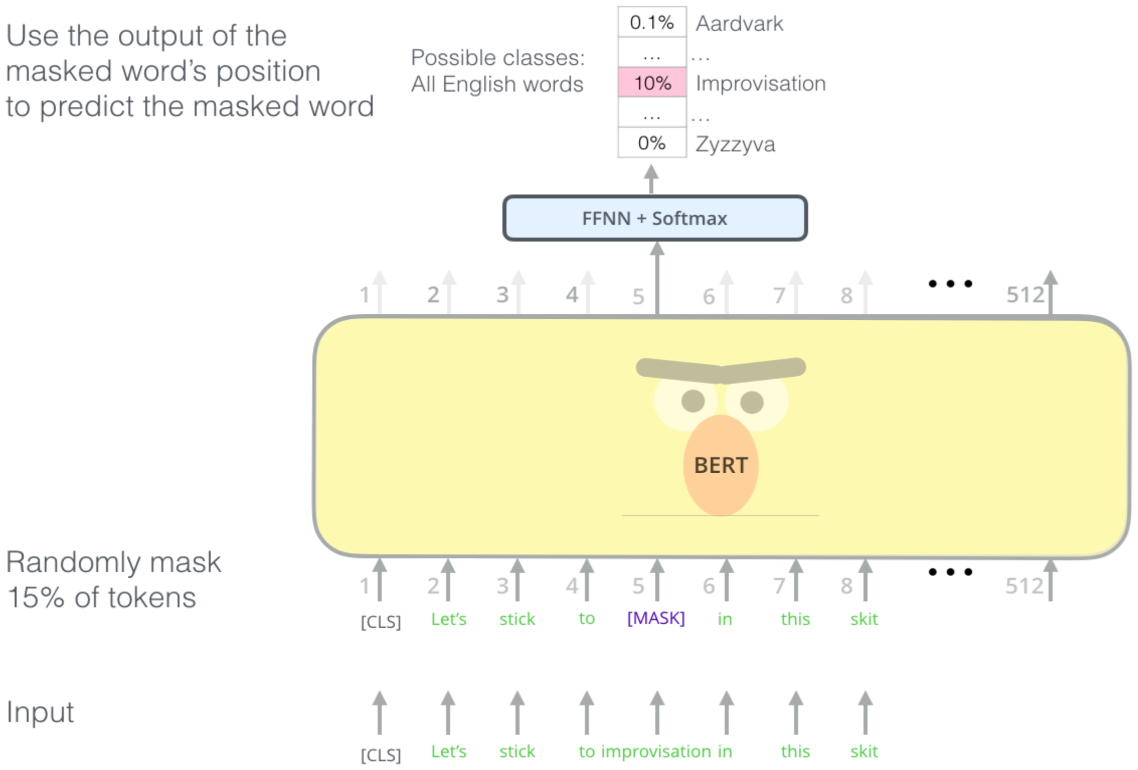
- Word embedding models learn a **single** representation per word by utilizing their context during training.
- Language models learn how each word interacts with others. The embeddings produced are **dependent** on the word itself and the way it is used in the sentence.

Language models

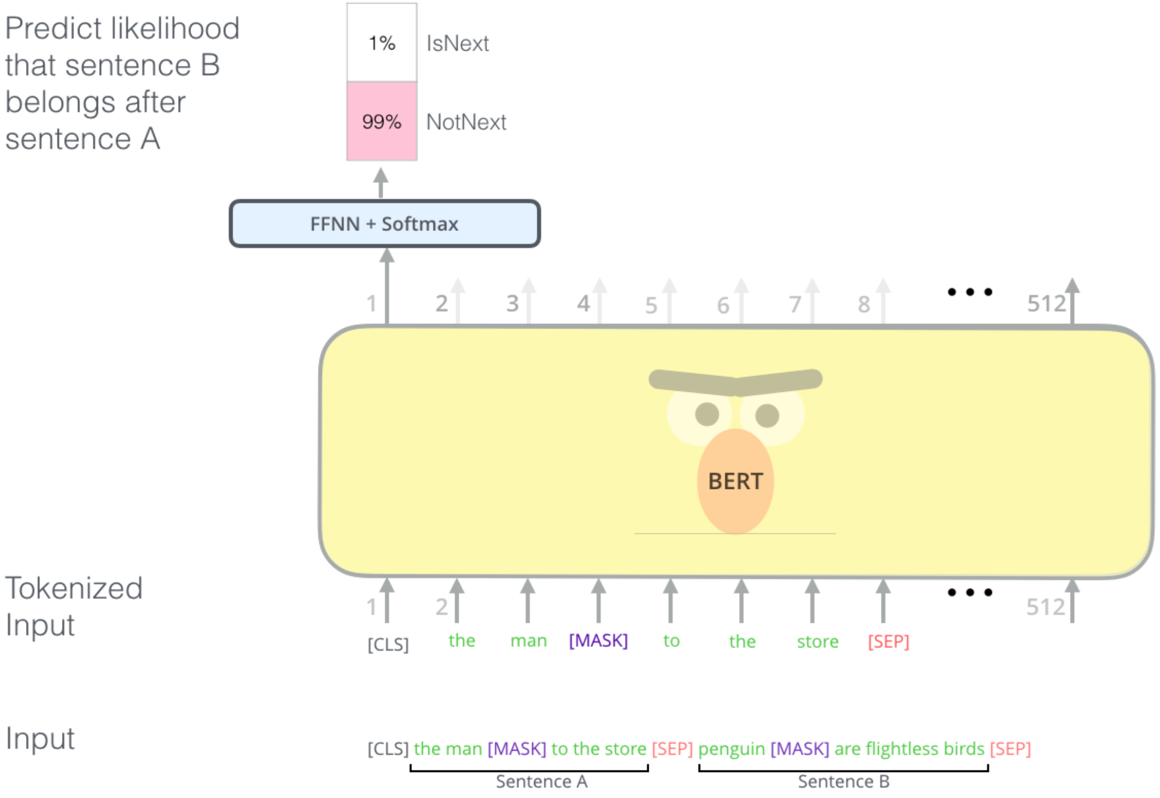
Bidirectional Encoder Representations from Transformers (***BERT***)

- BERT architecture: Multi-layer bidirectional Transformer encoder.
- ***BERT unsupervised pre-training*** consists of two tasks:
  - ***Mask Language Model*** finds the masked/hidden words by looking at their context.
  - ***Next Sentence Prediction*** predicts the appearance order two input sentences A, B.

Language models



Masked Language Model.

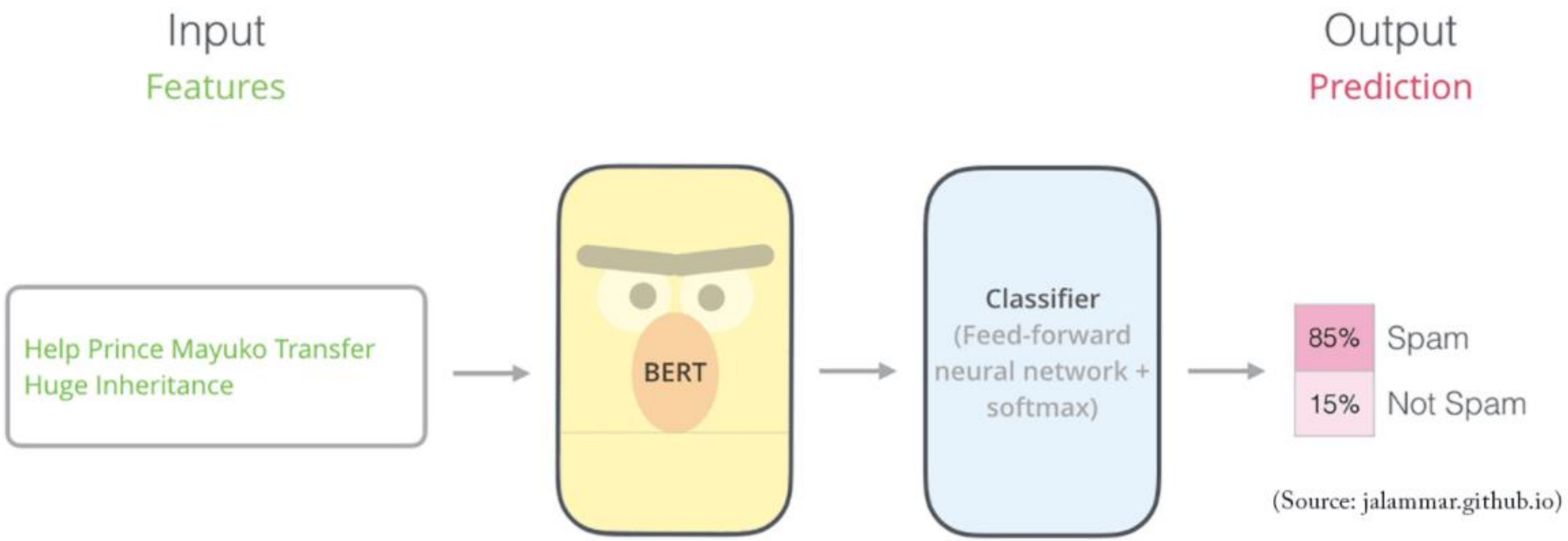


Next sentence prediction.

BERT pre-training

Artificial Intelligence & Information Analysis Lab

Language models



Bert Fine-tuning: supervised training on specific task.

Language models

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\mathrm{BERT_{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\mathrm{BERT_{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

BERT accuracy in different tasks.

Leaderboards

TREND	DATASET	BEST METHOD	PAPER TITLE
	SST-2 Binary classification	T5-3B	Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer
	SST-5 Fine-grained classification	RoBERTa-large+Self-Explaining	Self-Explaining Structures Improve NLP Models
	IMDb	NB-weighted-BON + dv-cosine	Sentiment Classification Using Document Embeddings Trained with Cosine Similarity
	Yelp Binary classification	BERT large	Unsupervised Data Augmentation for Consistency Training
	Yelp Fine-grained classification	BERT large	Unsupervised Data Augmentation for Consistency Training
	MR	byte mLSTM7	A La Carte Embedding: Cheap but Effective Induction of Semantic Feature Vectors
	Amazon Review Polarity	BERT large	Unsupervised Data Augmentation for Consistency Training
	Amazon Review Full	BERT large	Unsupervised Data Augmentation for Consistency Training
	SemEval 2014 Task 4 Subtask 1+2	GRACE	GRACE: Gradient Harmonized and Cascaded Labeling for Aspect-based Sentiment Analysis
	CR	Block-sparse LSTM	GPU Kernels for Block-Sparse Weights
	Multi-Domain Sentiment Dataset	Distributional Correspondence Indexing	Revisiting Distributional Correspondence Indexing: A Python Reimplementation and New Experiments
	MPQA	STM+TSED+PT+2L	The Pupil Has Become the Master: Teacher-Student Model-Based Word Embedding Distillation with Ensemble Learning
	DBRD	RobBERT v2	RobBERT: a Dutch RoBERTa-based Language Model
	Twitter	AEN-BERT	Attentional Encoder Network for Targeted Sentiment Classification

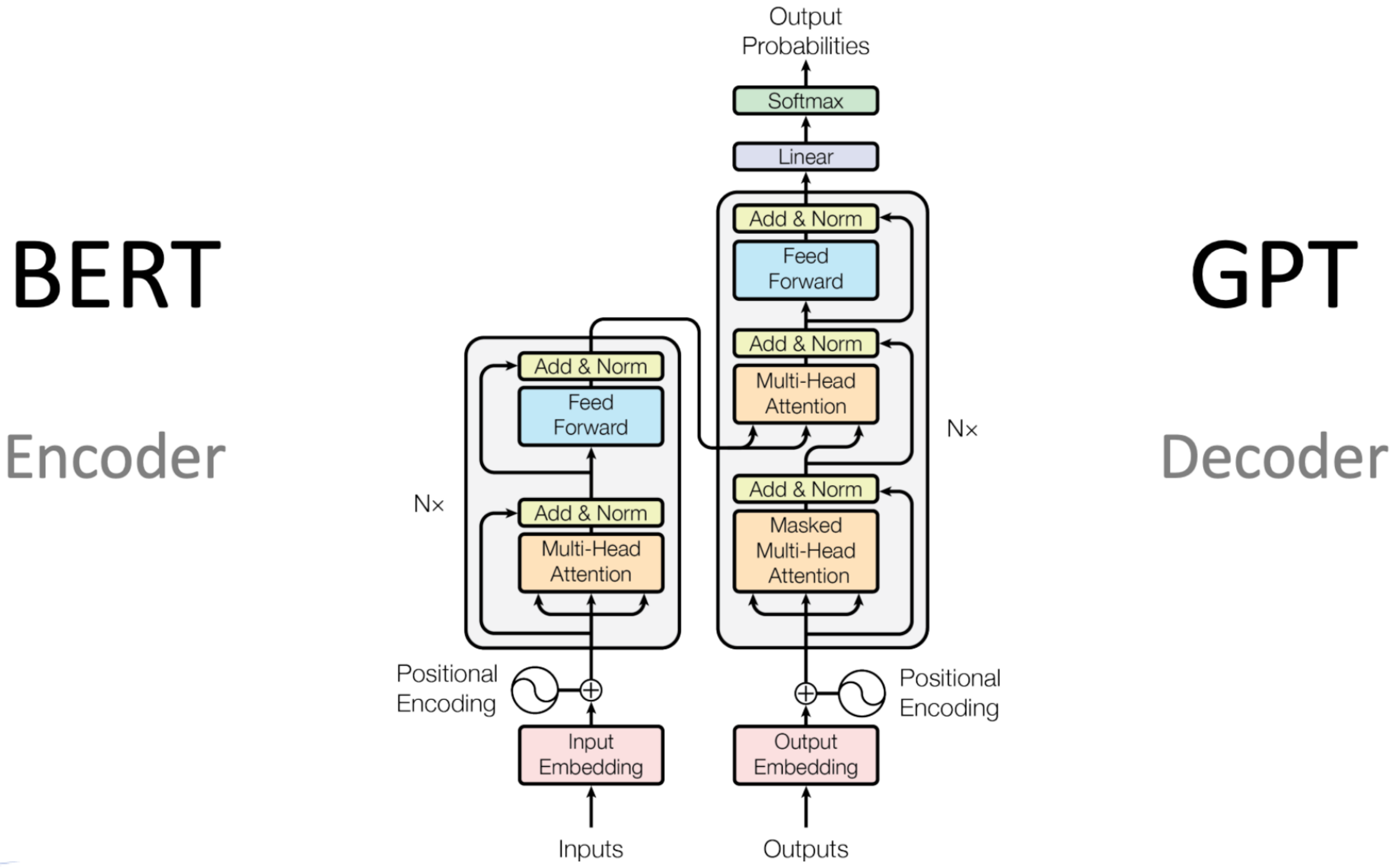
BERT vs GPT

BERT:

- Only encoder
- 2 pretraining objectives
- Bidirectional

GPT:

- Only decoder
- Fine tuning not always necessary
- Bigger pretraining corpus



Artificial Intelligence & Information Analysis Lab

BERT vs GPT

BERT is best at:

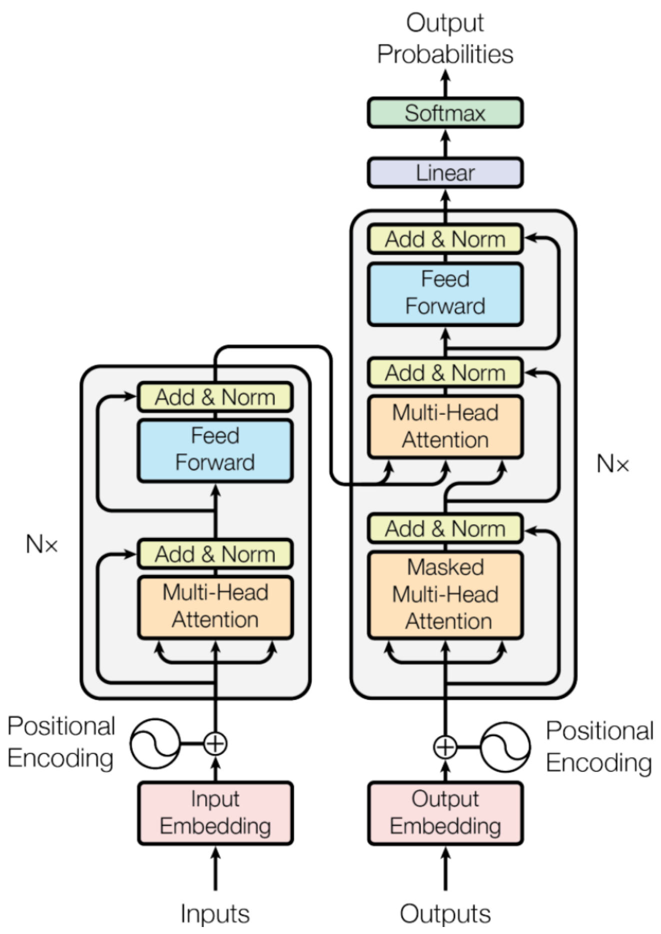
- Sentiment analysis
- Question Answering

GPT is best at:

- Text generation
- Summarization
- Translation

BERT

Encoder



GPT

Decoder

Chat-GPT

GPT models generate words based on the input and words already generated. *What if we train a GPT model on human conversation?*

- Gather prompts and desired output behavior.
- Get humans to rank outputs from best to worst (reward model).
- Create a policy based on the reward model.

Result: Chat-GPT

Bibliography

[TUR2009] Turing, Alan M. "Computing machinery and intelligence." *Parsing the turing test*. Springer, Dordrecht, 2009. 23-65.

[HUT2004]Hutchins, W. John. "The Georgetown-IBM experiment demonstrated in January 1954." *Conference of the Association for Machine Translation in the Americas*. Springer, Berlin, Heidelberg, 2004.

[WEI1966] Weizenbaum, Joseph. "ELIZA—a computer program for the study of natural language communication between man and machine." *Communications of the ACM* 9.1 (1966): 36-45.

[TAP2019] Tappert, Charles C. "Who is the father of deep learning?." *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2019.

[IVA1967] Ivakhnenko, Alekseĭ Grigor'evich, et al. *Cybernetics and forecasting techniques*. Vol. 8. American Elsevier Publishing Company, 1967.

[BEN2000] Bengio, Yoshua, Réjean Ducharme, and Pascal Vincent. "A neural probabilistic language model." *Advances in neural information processing systems* 13 (2000).

Bibliography

[COL2008] Collobert, Ronan, and Jason Weston. "A unified architecture for natural language processing: Deep neural networks with multitask learning." *Proceedings of the 25th international conference on Machine learning*. 2008.

[MIK2013] Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

[SUT2013] Sutskever, Ilya. *Training recurrent neural networks*. Toronto, ON, Canada: University of Toronto, 2013.

[KAL2014]Kalchbrenner, Nal, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." *arXiv preprint arXiv:1404.2188* (2014).

[ZHA2015] Zhang, Ye, and Byron Wallace. "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification." *arXiv preprint arXiv:1510.03820* (2015).

[SUT2014] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." *Advances in neural information processing systems* 27 (2014).

Bibliography

[BAH2014] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

[VAS2017] Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

[WES2014] Weston, Jason, Sumit Chopra, and Antoine Bordes. "Memory networks." *arXiv preprint arXiv:1410.3916* (2014).

[DAI2015] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." *Advances in neural information processing systems* 28 (2015).

[MCC2017] McCann, Bryan, et al. "Learned in translation: Contextualized word vectors." *Advances in neural information processing systems* 30 (2017).

[PET2018] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Bibliography

[RAD2018] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

[DEV2018] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[RUM1986] Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." *nature* 323.6088 (1986): 533-536.

[HOC1997] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.

[PEN2014] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global vectors for word representation." *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014.

[BOJ2017] Bojanowski, Piotr, et al. "Enriching word vectors with subword information." *Transactions of the association for computational linguistics* 5 (2017): 135-146.

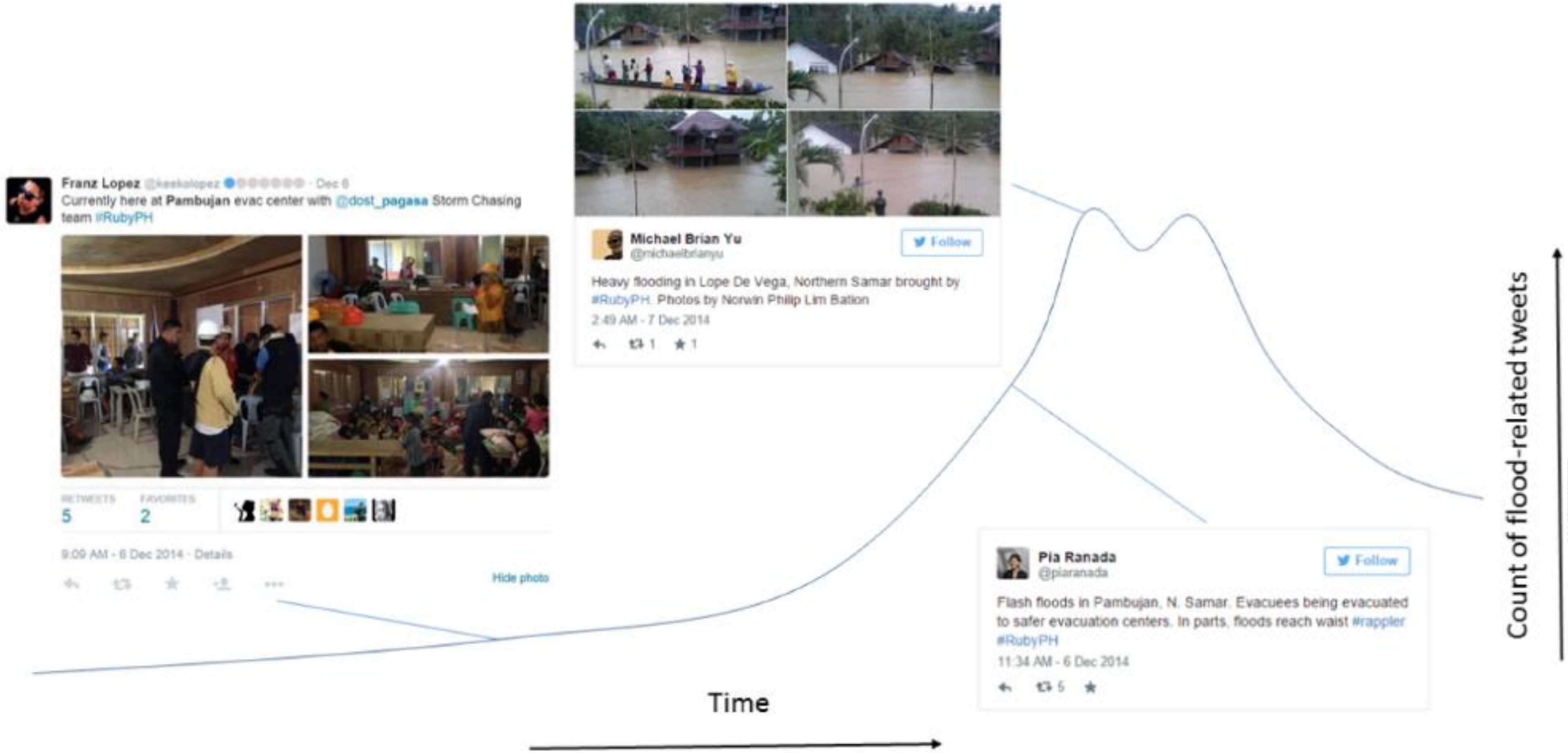
Bibliography

[RUD2018] S. Ruder, “A Review of the Neural History of Natural Language Processing”, https://ruder.io/a-review-of-the-recent-history-of-nlp/, 2018.

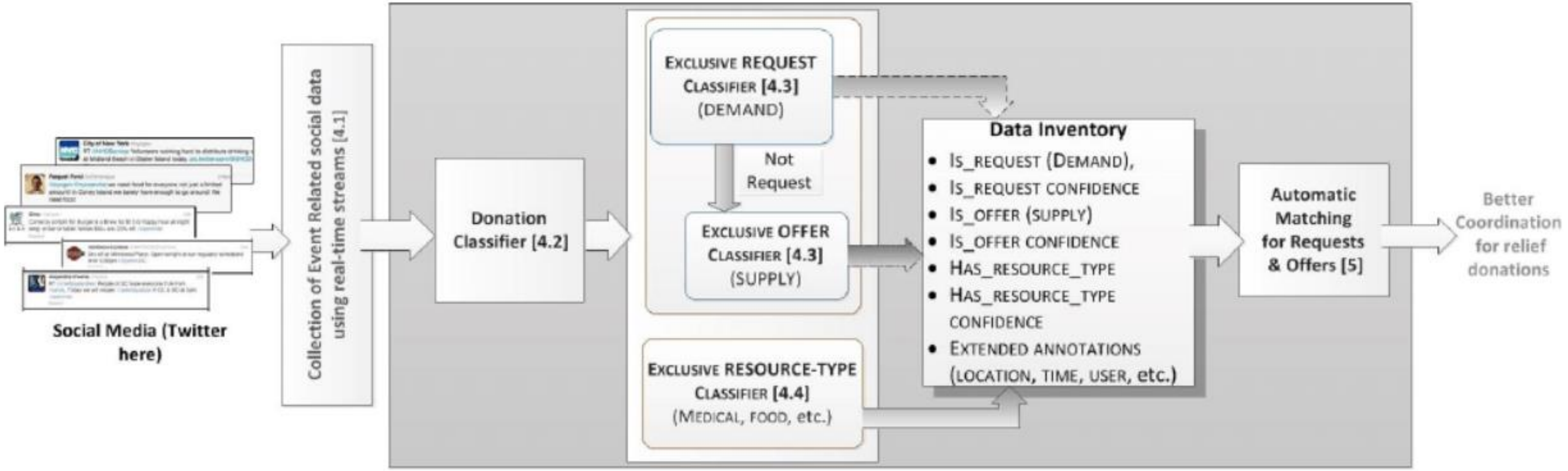
[THI2018] H. Thilakarathne, “One-Hot Encoding in Practice”, https://naadispeaks.wordpress.com/tag/one-hot-encoding/, 2018.

[VAS2017] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, "Attention is all you need“, Advances in Neural Information Processing Systems 30 (2017).

NLP in Natural Disasters



NLP in Natural Disasters



Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Polydoros Giannouris
polydoros@ece.auth.gr