Deep Learning for Text analysis

Jan Platos

2018-09-09

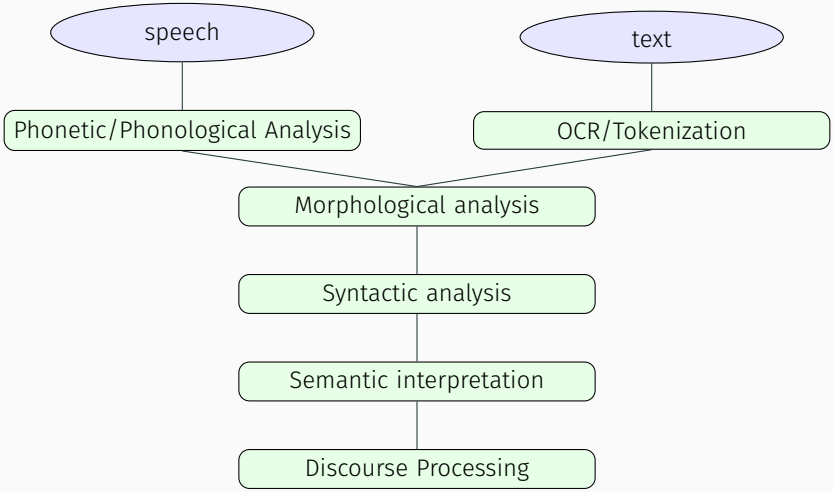
Table of Contents

Natural Language Processing

Natural Language Processing

- **Natural Language Processing** (NLP) is a research field at the intersection of
  - computer science
  - artificial intelligence
  - linguistics
- **Goal** is to process and understand natural Language in order to perform tasks that are useful, e.g.
  - Syntax checking
  - Language translation
  - Personal assistant (Siri, Google Assistant, Jarvis, Cortana, ...)
- **Note:** Fully understanding and representing the meaning of language is a difficult goal and is expected to be AI-complete.

Natural Language Processing



Natural Language Processing

- **Applications** of the NLP in a real life
  - Spell checking, keyword search, synonyms finding
  - Important data extraction from text (security codes, product prices, location, named entity, etc.)
  - Classification of content
  - Sentiment analysis
  - Topic extraction, topic evolution
  - Authorship identification, plagiarism detection
  - Machine translation
  - Dialog systems
  - Question answering system

Human Language Properties

- A human language is a system designed to transfer the meaning from speaker/writer to listener/reader.
- A human language uses an encoding that is simple for child to quickly learn and which changes during time.
- A human language is mostly discrete/symbolic/categorical signaling system.
  - Sounds
  - Gesture
  - Writing
  - Images
- The symbols are invariant across different encodings.

Deep learning in NLP - History

- Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition, Dahl et. al. 2012
  - A combined model of Hidden Markov Model, Deep Neural networks and Context dependency
  - Optimization on the GPU
  - Error reduction achieved is 32% with respect to traditional approaches.
- ImageNet Classification with Deep Convolutional Neural Networks, Krizhevsky, Sutskever, & Hinton, 2012
  - A model consist of Rectified Linear Units and Deep Convolution Networks.
  - Optimization on the GPU
  - Error reduction achieved is 37% with respect to traditional approaches.

Deep learning in NLP - Motivation

- NLP is HARD
  - Complexity in representation, learning and using linguistic/situation/contextual/word/visual knowledge.
  - Human languages are ambiguous:

Deep learning in NLP - Motivation

- NLP is HARD
  - Complexity in representation, learning and using linguistic/situation/contextual/word/visual knowledge.
  - Human languages are ambiguous:
    - I made her duck

Deep learning in NLP - Motivation

- NLP is HARD
  - Complexity in representation, learning and using linguistic/situation/contextual/word/visual knowledge.
  - Human languages are ambiguous:
    - **I made her duck**
    - I cooked waterfowl for her benefit (to eat)
    - I cooked waterfowl belonging to her
    - I created the (plaster?) duck she owns
    - I caused her to quickly lower her head or body
    - I waved my magic wand and turned her into undifferentiated waterfowl
- Deep models are know to be able to learn complex models
- The amount of data is huge as well as the amount of computational power

Deep learning in NLP - Applications

- Combination of Deep Learning with the goals and ideas of NLP

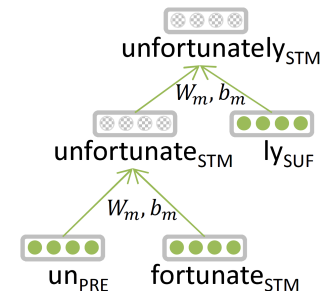
Deep learning in NLP - Applications

- Combination of Deep Learning with the goals and ideas of NLP
- **Word similarities** is a task to compute similarity between words to discover similarities without guiding (unsupervised learning)
- Nearest words for **FROG**:

1. frogs
2. toad
3. litoria (a king of frog)
4. leptodactylidae (the southern frogs form) ...

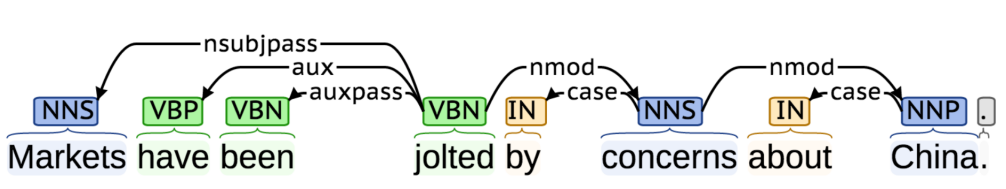
Deep learning in NLP - Applications

- Combination of Deep Learning with the goals and ideas of NLP
- **Word similarities** is a task to compute similarity between words to discover similarities without guiding (unsupervised learning)
- **Morphology** reconstruction and representation for improvement of word similarities.



Deep learning in NLP - Applications

- Combination of Deep Learning with the goals and ideas of NLP
- **Word similarities** is a task to compute similarity between words to discover similarities without guiding (unsupervised learning)
- **Morphology** reconstruction and representation for improvement of word similarities.
- **Sentence structure parsing** for precise grammatical structure identification.



Deep learning in NLP - Applications

- Combination of Deep Learning with the goals and ideas of NLP
- **Word similarities** is a task to compute similarity between words to discover similarities without guiding (unsupervised learning)
- **Morphology** reconstruction and representation for improvement of word similarities.
- **Sentence structure parsing** for precise grammatical structure identification.
- **Machine translation** now live in Google Translate, **Question Answering** system live in Google Assistant, Siri, etc.

Representation of the meaning of a word

Representation of the meaning of a word

- The **meaning** means:
  - the idea that is represented by a word, phrase, etc.
  - the idea that a person wants to express by using words, signs, etc.
  - the idea that is expressed in a work of writing, art, etc.
- A WordNet is a great resource of meaning:
  - A complex network of words made by human.
  - A list of synonyms, hypernyms (generalization), antonyms, etc.
  - A word category with dictionary-like description of a meaning.
  - A new meaning are missing in a database.
  - Some meaning and synonyms are valid only in some contexts.

Representation of the meaning of a word

- The standard representation is called **one-hot** vector.

$$\begin{aligned} motel &= [00000000100] \\ hotel &= [00000100000] \end{aligned}$$

- Vector dimension = number of word in a corpus
- Vectors are orthogonal $motel \cdot hotel = 0$
- Similarity cannot be defined on one/hot vector representation.
- WordNet may be used to extract synonyms for each word that will be used as similarity function, but ist too complicated approach.

Representation of the meaning of a word

A word’s meaning is given by the words that frequently appear close-by

Representation of the meaning of a word

A word's meaning is given by the words that frequently appear close-by

- When a word apears in the text, its context is set by the words that appear nearby (usually withing a fixed window).
- Many context windows for each word are used for representation of the word.

Representation of the meaning of a word

A word's meaning is given by the words that frequently appear close-by

- When a word apears in the text, its context is set by the words that appear nearby (usually withing a fixed window).
- Many context windows for each word are used for representation of the word.

Example:

...reasonable and to prevent the	network	trips from swamping out the execution...
...distance between nodes;	network	traffic or bandwidth constraints; ...
...beyond your control (i.e.	network	outage, hardware failure) or the latency ...
...experience was a temporarily-high	network	load which caused a timeout...
...is removed (i.e. temporary	network	disconnection resolved) then ...
...see their involvement with the	network	and its digital properties expand ...
...but cant get mobile	network	connection to work. Basically ...

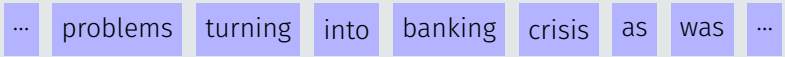
Word2vec framework

Word2vec is a framework for learning word vectors.

- We have a large corpus of text.
- Every word in a fixed vocabulary is represented by a vector.
- Go through each position t in the text, which has a center word c and context words o.
- Use the similarity of the word vectors for c and o to calculate the probability of o given c.
- Keep adjusting the word vectors to maximize the probability.

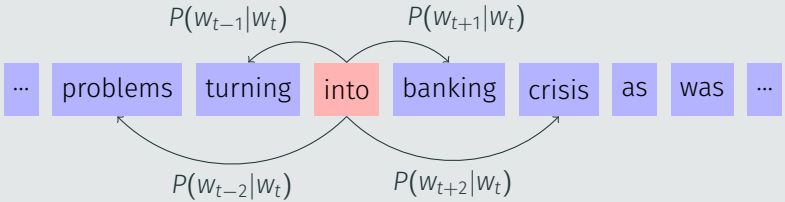
Word2vec framework

Example window and process of computing



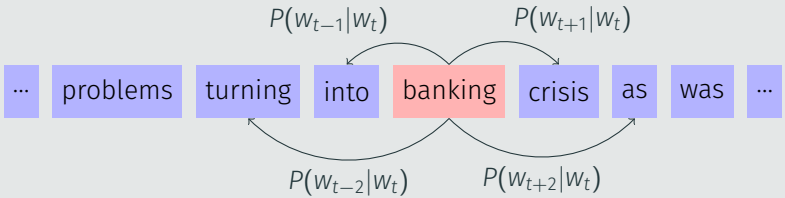
Word2vec framework

Example window and process of computing



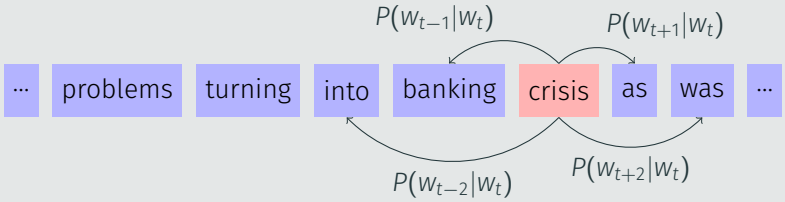
Word2vec framework

Example window and process of computing



Word2vec framework

Example window and process of computing



Word2vec framework - An objective function

- For each position $t = 1, \ldots, T$ predict context words within a window of fixed size m, given center word w_j.

$$Likelihood = L(\theta) = \prod_{t=1}^{T} \prod_{-m \leq j \leq m, j \neq 0} P\left(w_{t+j} | w_t; \theta\right)$$

- Where θ represents all variables to be optimized.
- The objective function (also cost or loss function) is defined as negative log likelihood:

$$J(\theta) = -\frac{1}{T} \log\left((L(theta)\right) = \sum_{t=1}^{T} \sum_{-m \leq j \leq m, j \neq 0} \log P\left(w_{t+j} | w_t; \theta\right)$$

- The minimization of the objective function will maximize the accuracy of the model.

Word2vec framework - An objective function

- The objective function need to be minimized:

$$J(\theta) = -\frac{1}{T}\log\left((L(theta)\right) = \sum_{t=1}^{T}\sum_{-m\leq j\leq m, j\neq 0}\log P\left(w_{t+j}|w_t;\theta\right)$$

- The calculation of the $P\left(w_{t+j}|w_t;\theta\right)$ is crucial.
- For each word w we use two vectors:
  - v_w when w is a center word.
  - u_w when the w is context word.
- For center word c and context word o the probability:

$$P(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in V}\exp\left(u_w^T v_c\right)}$$

Word2vec framework - A prediction function

$$P(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w \in V} \exp\left(u_w^T v_c\right)}$$

- $u_o^T v_c$ is a dot product that compares similarity of o and c (cosine similarity)
- $\sum_{w \in V} \exp\left(u_w^T v_c\right)$ normalize over the entire vocabulary V.
- It is an example of the **softmax** function $\mathcal{R}^n \rightarrow \mathcal{R}^n$.

$$softmax(x_i) = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)} = p_i$$

- The softmax function distribution maps arbitrary values of x_i to a probability distribution p_i
  - **max** because amplifies probability to largest x_i
  - **soft** because still assigns some probability to smaller x_i

Word2vec framework - Training a model

- The θ represents **all** model parameters, in one large vector.
- The vector has d-dimensional vectors and V-many words.

$$\theta = \begin{bmatrix} v_a \\ \vdots \\ v_z \\ u_a \\ \vdots \\ u_z \end{bmatrix} \in \mathcal{R}_{2dV}$$

- These parameters are then optimized.
- A **Gradient Descent** algorithm fits as well as **Stochastic Gradient Descent**.

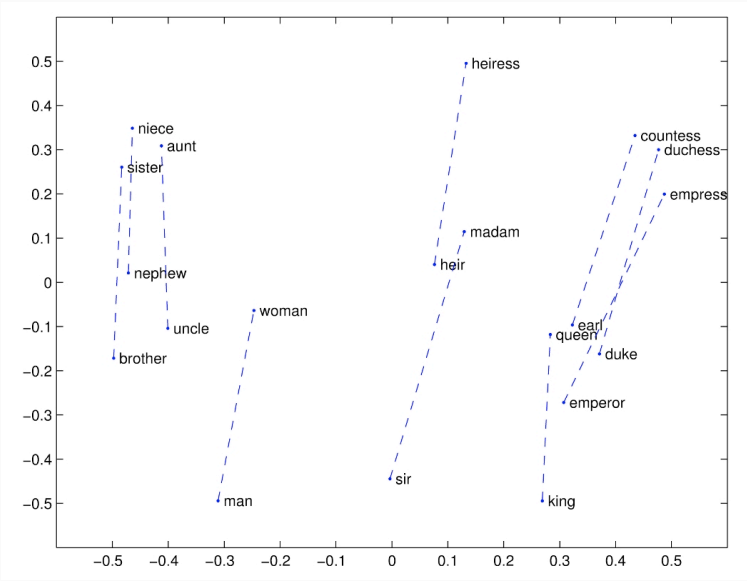
Word2vec framework - Variants

- Two base models are used:
  1. **Skip-Gram (SG)** where the contexts predicts words given the center word independently on position.
  2. **Continuous Bag of Words (CBOW)** where the center word is predicted from context words.
- Latent Semantics Analysis
  - A different approach that computes the similarity according to co-occurrence of words in a corpora.
  - Space requirements are enormous.
  - Incorporate Singular Value Calculation as a best approximation.
- GloVe: Global Vectors for Word Representation
  - Combines both techniques and defines modified objective function:

$$J(\theta) = \frac{1}{2} \sum_{i,j=1}^{W} f(P_{ij}) \left(u_i^T v_j - \log P_{ij} \right)^2$$

  - Fast training, scalable to huge corpora but works even on small ones.

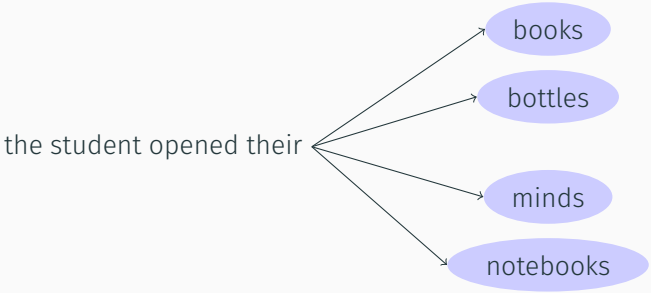
Glove Results



Language Modeling

Language Modeling

- Language modeling is a task of predicting what word comes next.



Language Modeling

- Language modeling is a task of predicting what word comes next.
- Given a sequence of words $x_1, x_2, \ldots, x_t$, compute the probability distribution of the next word x_{t+1}:

$$P(x_{t+1} = w_j | x_t, \ldots, x_1)$$

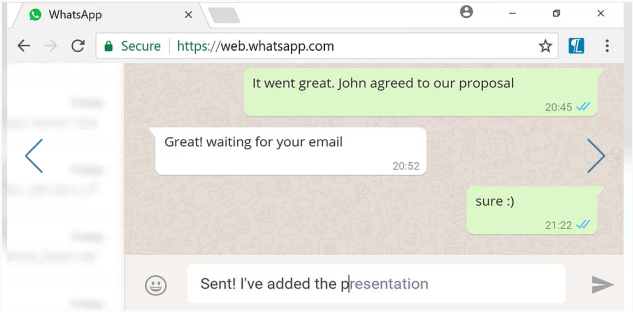
- Where w_j is a word in the vocabulary $V = \{w_1, \ldots, w_{|V|}\}$.

Language Modeling

- Language modeling is a task of predicting what word comes next.
- Given a sequence of words $x_1, x_2, \ldots, x_t$, compute the probability distribution of the next word x_{t+1}:

$$P(x_{t+1} = w_j | x_t, \ldots, x_1)$$

- Where w_j is a word in the vocabulary $V = \{w_1, \ldots, w_{|V|}\}$.

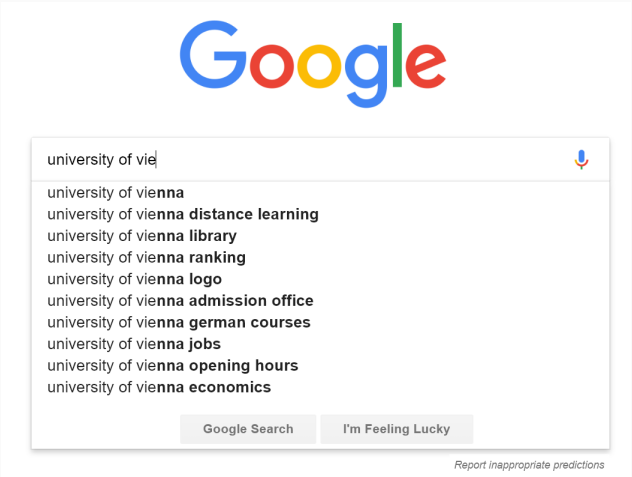


Language Modeling

- Language modeling is a task of predicting what word comes next.
- Given a sequence of words $x_1, x_2, \ldots, x_t$, compute the probability distribution of the next word x_{t+1}:

$$P(x_{t+1} = w_j | x_t, \ldots, x_1)$$

- Where w_j is a word in the vocabulary $V = \{w_1, \ldots, w_{|V|}\}$.



n-Gram Language model

- An **n-gram** is a chunk of n consecutive words:
  - **uni**grams: "the", "students", "opened", "their"
  - **bi**grams: "the students", "students opened", "opened their"
  - **tri**grams: "the students opened", "students opened their"
  - **4-**grams: "the students opened their"
- Idea is to collect a statistics about how frequently different n-grams are and use them to predict next word.
- We assume that a word x_{t+1} depends only on the preceding $(n-1)$ words.

$$P(x_{t+1} = w_j | x_t, \ldots, x_{t-n+2}) = \frac{P(x_{t+1}, x_t, \ldots, x_{t-n+2})}{P(x_t, \ldots, x_{t-n+2})}$$

- The values may be computed from the corpora.

n-Gram Language model

- The language model may be used to generate text.

today the ...

n-Gram Language model

- The language model may be used to generate text.

today the price ...

n-Gram Language model

- The language model may be used to generate text.

today **the price** ...

n-Gram Language model

- The language model may be used to generate text.

today **the price** of ...

n-Gram Language model

- The language model may be used to generate text.

today the **price of** ...

n-Gram Language model

- The language model may be used to generate text.

today the **price of** gold ...

n-Gram Language model

- The language model may be used to generate text.

today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share

n-Gram Language model

- The language model may be used to generate text.

today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share

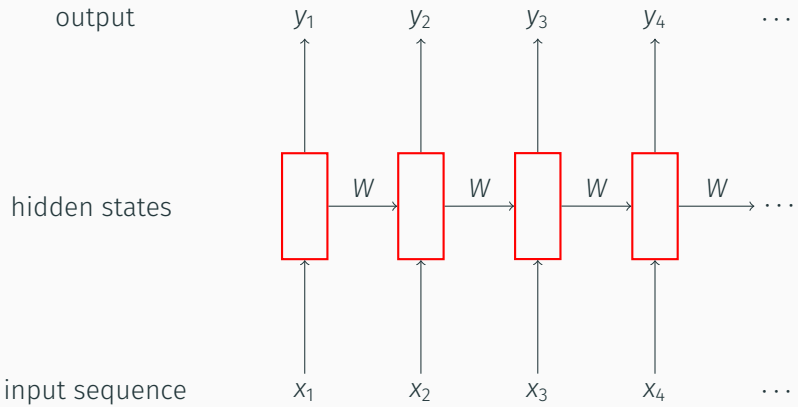
- The result is incoherent, more than two word need to be taken into account!!!
- The increasing of n leads to sparsity problem and increase the model size.
- Sparsity problem - the sequence never appear in the data.

Neural Language model

- The task:
  - Input: sequence of words: $x_1, \ldots, x_t$
  - Output: Probability of next word $P(x_{t+1} = w_j | x_t, \ldots, x_1)$
- A window approach may work similarly as for n-grams.
  1. Input is one-hot-vectors
  2. Compute word embedding for each word and concatenate as input.
  3. Define a hidden layer.
  4. Set output as **softmax** function over the hidden layer.
- This solves the problem of sparsity and reduces size of the model to linear.
- Some problems remains:
  - The fixed window limits the precision and is never large enough.
  - The weights are not shared between words in a window.

Recurrent Neural Network (RNN)

- A neural network that is able to incorporate unlimited input.



Recurrent Neural Network (RNN)

- Advantages:
  - Can process any length of input.
  - Model size does not increasing with the input length.
  - Computation of current step can use information from many steps back.
  - Weights are shared across time steps.
- Disadvantages:
  - Computation is very slow.
  - It is difficult in practice access information from many steps back.

Long Short Term memories (LSTM)

- More complex version of RNN.
- Capable to learn long term dependencies practically.
- Multi-layer architecture, with shortcuts and adaptive learning.
- The ”knowledge” flow is regulated using Gates.
- Gates are non-linear neural net layer (sigmoid) and regulate the amount of information that is let through.
- It solves the problem with long term memories, while maintain short term memories too.

Recurrent Neural Network (RNN) - Examples

RNN as a political speech writer (input phrase Jobs)[1]

Good afternoon. God bless you.
The United States will step up to the cost of a new challenges of the American people that will share the fact that we created the problem. They were attacked and so that they have to say that all the task of the final days of war that I will not be able to get this done. The promise of the men and women who were still going to take out the fact that the American people have fought to make sure that they have to be able to protect our part. ...

[1] https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0

Recurrent Neural Network (RNN) - Examples

LSTM as a novelist[2]

”The Malfoys!” said Hermione.
Harry was watching him. He looked like Madame Maxime.
When she strode up the wrong staircase to visit himself.
”I’m afraif I’ve definitely been suspended from power, no chance - indeed?” said Snape. He put his head back behind them and read groups as they crossed a corner and fluttered down onto their ink lamp, and picked up his spoon. The doorbell rang. It was a lot cleaner down in London...

[2] https://medium.com/deep-writing/harry-potter-written-by-artificial-intelligence-8a9431803da6

Language models - usability

- Language modelling is a sub-component of other NLP systems:
  - Speech Recognition
    - An LM generates transcription according to the audio.
  - Machine translation
    - An LM generate translation according to the original text.
  - Summarization
    - An LM generate summary conditioned on original text.

Neural Machine Translation

Neural Machine Translation

- Machine Translation is a task to translate sequence X from source language into sequence Y in target language.
- Historically (since 1950) rule-based models with bilingual dictionaries (mostly Russian to English).
- Since 1990 a probabilistic model extracted from data was used.
  - Searching for best sentence in English given the sequence in French

$$argmax_y P(y|x)$$

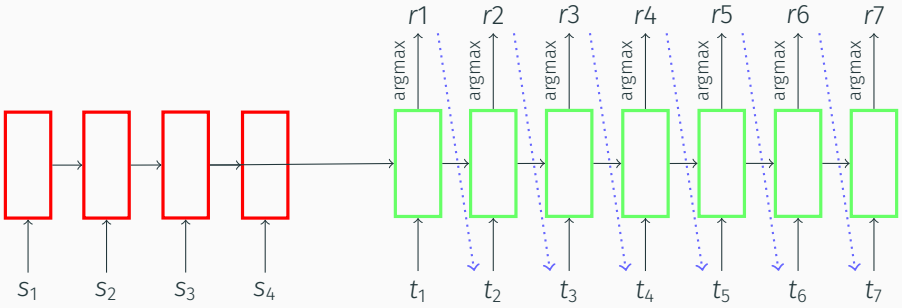
  - Bayes rule break this into two components that are learnt separately.

$$= argmax_y P(x|y)\, P(y)$$

  - $P(y)$ is a language model, $P(x|y)$ is a translation model.
  - $P(y)$ is learnt from monolingual data of good English text.
  - $P(x|y)$ is learnt from parallel corpus.

Neural Machine Translation

- Neural Machine Translation (NMT) is a way to do Machine Translation with a single neural network.
- The architecture is called sequence-to-sequence (seq2seq) and it involves two RNNs.



Neural Machine Translation

- Advantages
  - Better performance, more fluent, better context, better phrase similarities.
  - Its a single neural network that is optimized together at once.
  - Requires much less human engineering effort (no feature selection, the process is the same for all languages pairs).
- Disadvantages
  - Less interpretable (impossible to Debug the learning).
  - Difficult to control (no rules, guidance, etc.).
- Advancements
  - 2014 - first paper about NMT and seq2seq published.
  - 2016 - Google Translate switched into NMT.

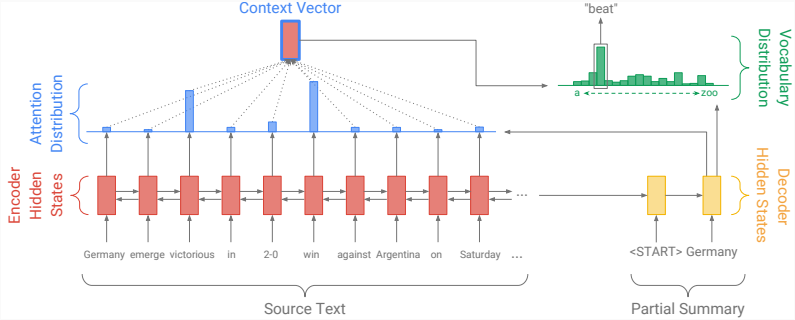
Neural Machine Translation - Improvements

- Attention
  - Idea: on each step of the decoded, focus on a particular part of the source sequence.
  - The attention information is used for output generation directly.
  - The attention highlight more important part of the source.
  - Improves the long term memory usability.
  - Applicable to other architectures than seq2seq.
- Usage:
  - Summarization (long text to short text)
  - Code generation (natural language into python script)

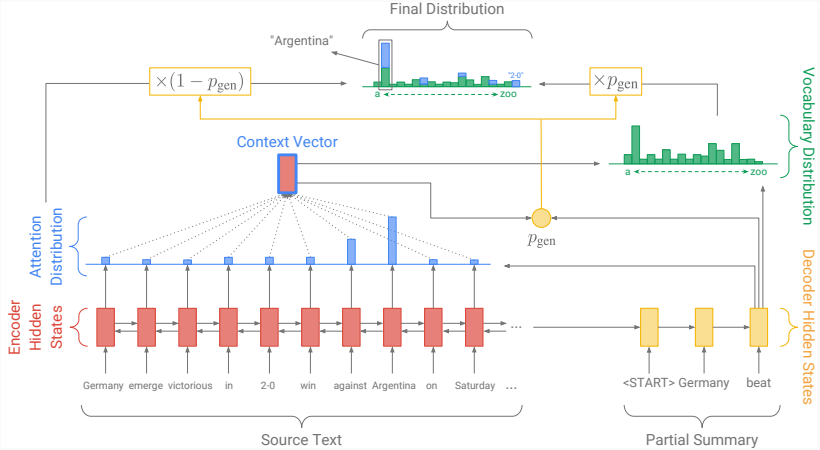
Seq2seq Example - Summarization

- Get To The Point: Summarization with Pointer-Generator Networks, A.See (Stanford), P.J. Liu (Google), Ch. D. Mannign (Stanford), 2016.
- Combination of :
  - Seq2seq attention model - the encoder (bidirectional LSTM) and decoder (unidirectional LSTM) cooperates with attention modeling mechanism.
  - Pointer generator network - a principle that is able to copy word directly from source text in case of words that are not in a vocabulary (names, locations, etc).
  - Coverage mechanism that remove repetitions in generated abstract.
- Training data - CNN/Daily mail dataset
  - News articles (781 tokens on average)
  - Multi-sentence summaries (56 tokens in average)
  - 287,226 training pairs
  - 13,368 validation pairs
  - 11,490 test pairs

Seq2seq Example - Summarization



Seq2seq Example - Summarization



Seq2seq Example - Summarization

- 256-dimensional hidden states
- 128-dimensional word embedding
- 21,499,600 parameters to optimized
- Tesla K40m GPU, batch size 16.
- 230,000 training iterations
- Training time was 3 days and 4 hours.

Seq2seq Example - Summarization

- 256-dimensional hidden states
- 128-dimensional word embedding
- 21,499,600 parameters to optimized
- Tesla K40m GPU, batch size 16.
- 230,000 training iterations
- Training time was 3 days and 4 hours.

Article: andy murray (...) is into the semi-finals of the miami open , but not before getting a scare from 21 year-old austrian dominic thiem, who pushed him to 4-4 in the second set before going down 3-6 6-4, 6-1 in an hour and three quarters. (...)

Summary: andy murray defeated dominic thiem 3-6 6-4, 6-1 in an hour and three quarters.

Seq2seq Example - Summarization

- 256-dimensional hidden states
- 128-dimensional word embedding
- 21,499,600 parameters to optimized
- Tesla K40m GPU, batch size 16.
- 230,000 training iterations
- Training time was 3 days and 4 hours.

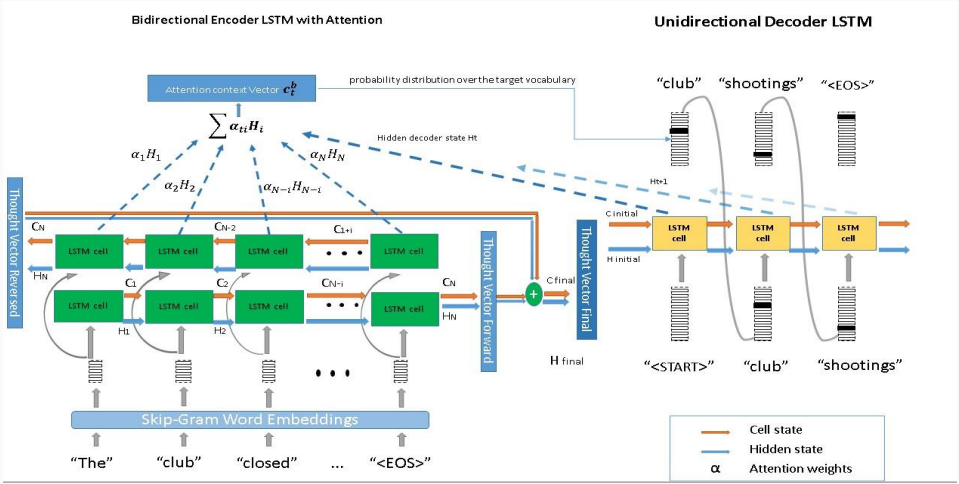
Article: (...) wayne rooney smashes home during manchester united 's 3-1 win over aston villa on saturday. (...)

Summary: manchester united beat aston villa 3-1 at old trafford on saturday..

Seq2seq Example - Summarization 2

- A work of Moseli Mots'oehli, University of Pretoria and me.
- Simplification of a model of See et. al.
- Encoder-Decoder Bidirectional LSTM architecture with Word2Vec for word embedding on source and one-hot encoding on target and Attention principle.

Seq2seq Example - Summarization 2



Seq2seq Example - Summarization 2

Article: usain bolt rounded off the world championships sunday by claiming his third gold in moscow as he anchored jamaica to victory in the mens 100 m relay. ...the british quartet, who were initially fourth, were promoted to the bronze which eluded their mens team. fraser pryce, like bolt aged , became the first woman to achieve three golds in the and the relay.

Golden Summary: usain bolt wins third gold of world championship. anchors jamaica to 100m relay victory. eighth gold at the championships for bolt. jamaica double up in womens 100m relay.

Summary: usain *usain* bolt wins third gold world championship anchors *anchors* jamaica x *x* relay victory *victory* eighth gold at bolt.

Seq2seq Example - Summarization 2

Article: it is official american president barack obama wants lawmakers to weigh in on whether to use military force in syria obama sent a letter to the heads of the house and senate on saturday night hours after announcing that he believes military action against syrian targets is the right step to take over the alleged use of chemical weapons the proposed legislation from obama asks congress to approve the use of military force ”to deter disrupt prevent and degrade the potential for future uses of chemical weapons or other weapons of mass destruction ...

Golden Summary: syrian official obama climbed to the top of the tree ”does not know how to get down” obama sends a letter to the heads of the house and senate obama to seek congressional approval on military action against syria aim is to determine whether

Summary: a syrian official *official* climbed *climbed* the top the *the* tree does *does* not *not not* obama get not sends

Seq2seq Example - Summarization 2

Article: with the sweltering summer bidding adieu and pleasant autumn temperatures setting in nows the time to explore new delhi travelers to the indian capital may hesitate to try the citys famed street foods fearing the notorious ”delhi belly ” but skip the street food scene and you miss an essential part of the delhi experience here are seven street delicacies among delhis endless choices including a mix of vegetarian non veg and dessert ...

Golden Summary: if you have not tried these street foods you have not been to delhi the most iconic chaat are aloo tikki dahi bhalla and papri chaat the best kulfi ice cream is topped with rose milk faluda

Summary: new if you *you* have not *not* foods you have *have have* not been delhi to the most *most* is

References

References

1. CS224n: Natural Language Processing with Deep Learning, Stanford class, http://web.stanford.edu/class/cs224n/index.html
2. Get To The Point: Summarization with Pointer-Generator Networks, Abigail See, Peter J. Liu, Christopher D. Manning, 2016, https://nlp.stanford.edu/pubs/see2017get.pdf
3. Bidirectiona-LSTM-for-text-summarization-, Moseli Motsoehli, https://github.com/DeepsMoseli/Bidirectiona-LSTM-for-text-summarization-
4. Better Word Representations with Recursive Neural Networks for Morphology, Minh-Thang Luong, Richard Socher, and Christopher D. Manning, 2013 https://nlp.stanford.edu/~lmthang/data/papers/conll13_morpho.pdf
5. Speech and Language Processing (3rd ed. draft), Dan Jurafsky and James H. Martin, https://web.stanford.edu/~jurafsky/slp3/
6. ...

Thank you for your attention