Introduction to NLP

Bert included ;)

Nicholas Vachon

NIVELOP

Table des matières

History

Rule Based Methods (1950 until late ‘80)

- Many language-processing systems were designed by symbolic methods:
  - hand-coding a set of rules
  - dictionary lookup
- Drawbacks:
  - Increase accuracy -> Increase rules -> Hard to manage
  - Treatment of rare cases or error hard
- Continue to be relevant for contexts in which interpretability and transparency is required or in low data regimes.

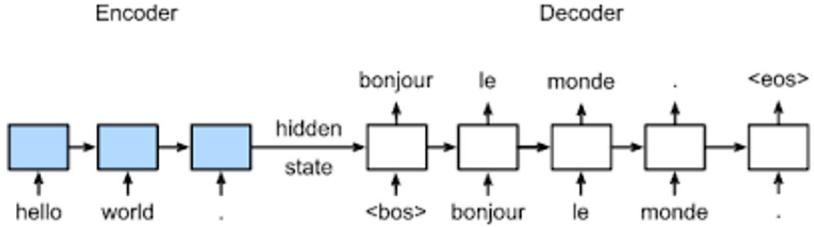
History

Statistical Methods (late 1980 – 2010)

- Usage of machine-learning
  - Using *statistical inference* to automatically learn rules through the analysis of large corpora
  - First: hard decision (e.g. tree based models)
  - Then statistical models which make soft, probabilistic decisions
    - Better with unfamiliar input, errors
    - More reliable results when integrated into a larger system
- Still used when *statistical* interpretability and transparency is required
- Drawback:
  - Require elaborate feature engineering

History

Neural Networks (2010 +)



- Semantic representation
  - Feature engineering -> word embeddings
- End-to-end learning
  - From embedding to higher level tasks (e.g. sentiment analysis, question answering)
  - Stop relying on pipeline of separate intermediate tasks (e.g., part-of-speech tagging)
- NMT uses sequence-to-sequence techniques
  - Obviates need for intermediate (e.g. word alignment, language modeling) used in statistical machine translation (SMT)

Words as numbers

Words as numbers

Words as numbers is equivalent to feature engineering

> In classical ML we use **hidden properties** to regress / classify on. For text : we rely on the **underlying** semantic and properties of the text.

> How to measure **semantic** and **properties** is heavily dependent of the task to accomplish (lowering, stemming, should not be applied to all tasks), but it basically pertains to feature engineering for texts.

> Derive **predictable** and **analyzable** features from textual data.

- Analyzable : for which mathematical **operators** "make sense" (regarding the task)
  - Ex : lowering and text classification : lowering does not tamper with the semantic of the text.

Words as Numbers

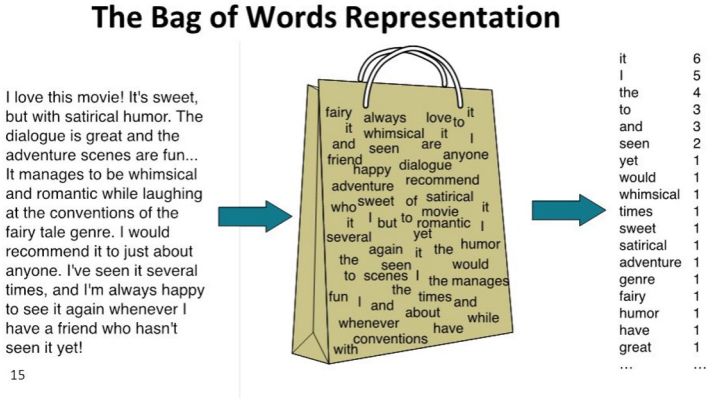
Examples of textual representations

- Sequential Numbers?
- If: to = 1, be = 2, or =3, not = 4
  - « To be or not to be » = (1,2,3,4,1,2)
  - Problems: size of values and their ordering have no significance

    -> not working for stats (hence not for most ML), but useful to compute to estimate the similarity of two texts (but not efficient)

Words as Numbers

Bag of Words



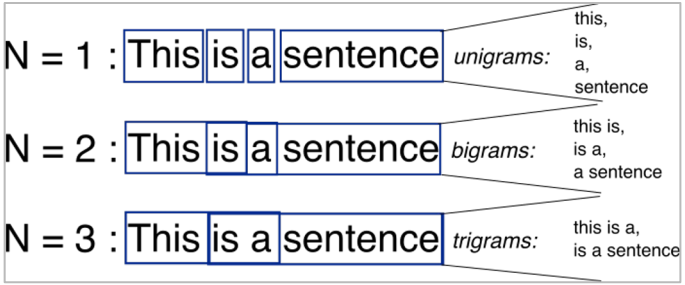
- Representing text as set of words
  - « To be or not to be » = {'to'=2, 'be'=2, 'or'=1, 'not'=1}
- One-hot encoding
  - Size of vocabulary is size of vector
  - If: to = (1,0,0,0), be = (0,1,0,0), or = (0,0,1,0), not = (0,0,0,1)
  - « To be or not to be » = (2,2,1,1) -> Bag of words representation
- Surprisingly good for ML even with lost of sequencing and semantic
- Drawback: sparsity, high dimensionality, no context nor semantics (-> don't use random forest)

$$Vocab = \begin{bmatrix} \text{I} \\ \text{love} \\ \text{like} \\ \text{cats} \\ \text{dogs} \end{bmatrix}$$

$$v(\text{I}) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, v(\text{like}) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, v(\text{dogs}) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Words as Numbers

n-grams



- Representing text as set of *combined* words
- n = how many words combined
- Has better ‘spatial’ representation the BoW
- BoW is n-grams with n=1
- Bigrams (i.e. 2-gram) of « To be or not to be »:
  - ‘to be’ ‘be or’ ‘or not’ ‘not to’ ‘to be’ (4 items since ‘to be’ twice)
  - One hot encoding
    - Size is number of possible n words combinations
    - (2, 1, 1, 1) if order above followed
- Drawback: sparsity and high dimensionality (more than BoW)

Words as Numbers

Tf-idf

$$w_{x,y} = tf_{x,y} \times \log\left(\frac{N}{df_x}\right)$$

TF-IDF

Term x within document y

$tf_{x,y}$ = frequency of x in y

df_x = number of documents containing x

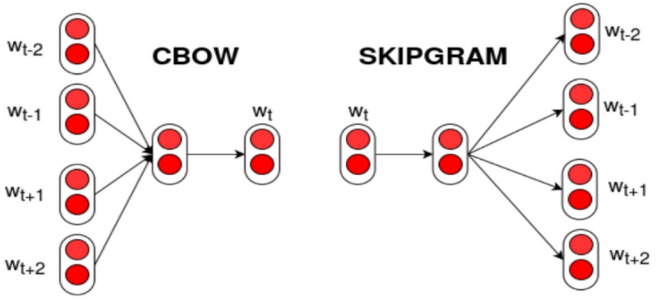
N = total number of documents

- BoW and n-gram problem:
  - All words have same importance (eg: ‘the’ same weight as ‘anthropomorphize’)
- Rescales the frequency of words by how often they appear in *all* texts
  - Frequent words like “the”, that are also frequent across other texts, get penalized
- Term Frequency — Inverse Document Frequency (tf-idf)
  - Improve BoW with weights
- Drawback: sparsity, high dimensionality

Words as Numbers

Word2Vec

- Small, dense, with context
- Unsupervised
- Cbow fast, skip-gram better in general
- Simple math operations
  - ‘king’ – ‘man’ + ‘woman’ = ‘queen’
- Once training completed, same embedding (representation) whatever the context of the word (not BERT). Can disambiguate with different techniques (e.g. part-of-speech).



	Words			
Dimensions	King	Queen	Woman	Princess
Royalty	0.990	0.990	0.020	0.980
Masculinity	0.990	0.050	0.010	0.020
Femininity	0.050	0.930	0.999	0.960
Age	0.700	0.600	0.500	0.100

Word as Numbers

Word Embedding – Other Techniques

- Dimensionality Reduction
- Co-occurrence Matrix [1]
- probabilistic models [2]
- explainable knowledge base method [3]
- and explicit representation in terms of the context in which words appear [4]

[1] *Lebret, Rémi; Collobert, Ronan (2013). "Word Emdeddings through Hellinger PCA". Conference of the European Chapter of the Association for Computational Linguistics (EACL).2014*

[2]*Globerson, Amir (2007). "Euclidean Embedding of Co-occurrence Data"(PDF). Journal of Machine Learning Research.*

[3] *Qureshi, M. Atif; Greene, Derek (2018-06-04). "EVE: explainable vector based embedding technique using Wikipedia". Journal of Intelligent Information Systems. 53:137–165.*

[4] *Levy, Omer; Goldberg, Yoav (2014). Linguistic Regularities in Sparse and Explicit Word Representations (PDF). CoNLL. pp. 171–180.*

Models

Models

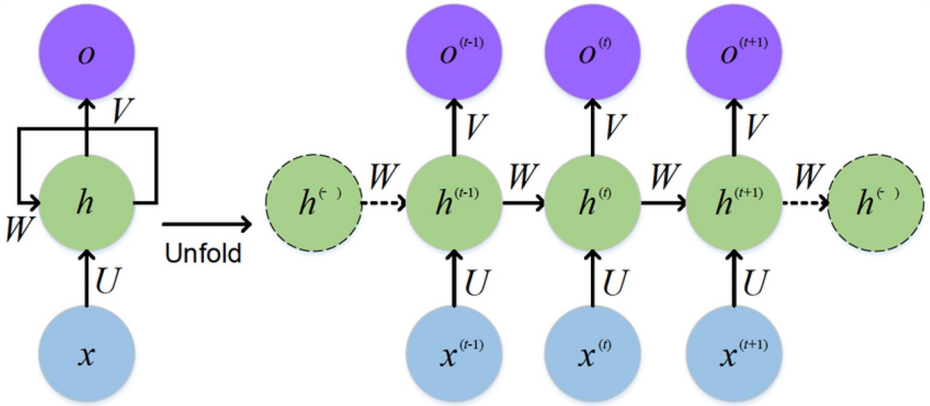
Sentence: A sequence of words

- *Sentence* with fixed length representation
  - BoW or tf-idf representation of sentences are fixed length (~vocabulary size)
  - Good with traditional ML models
- *Words* with fixed length representation
  - Word embeddings have a fixed length
  - But, sentence size now varies => need sequential models
- Sequential models take variable size of input and output
  - Recurrent Neural Networks (RNN)
  - Bert (not sequential, but can act as is)

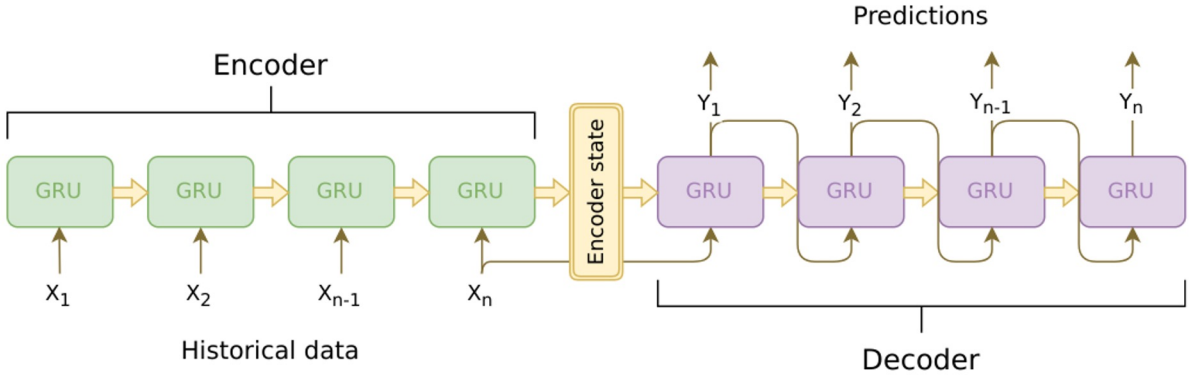
Models

RNN

Basic RNN



Sequence-to-sequence GRU model

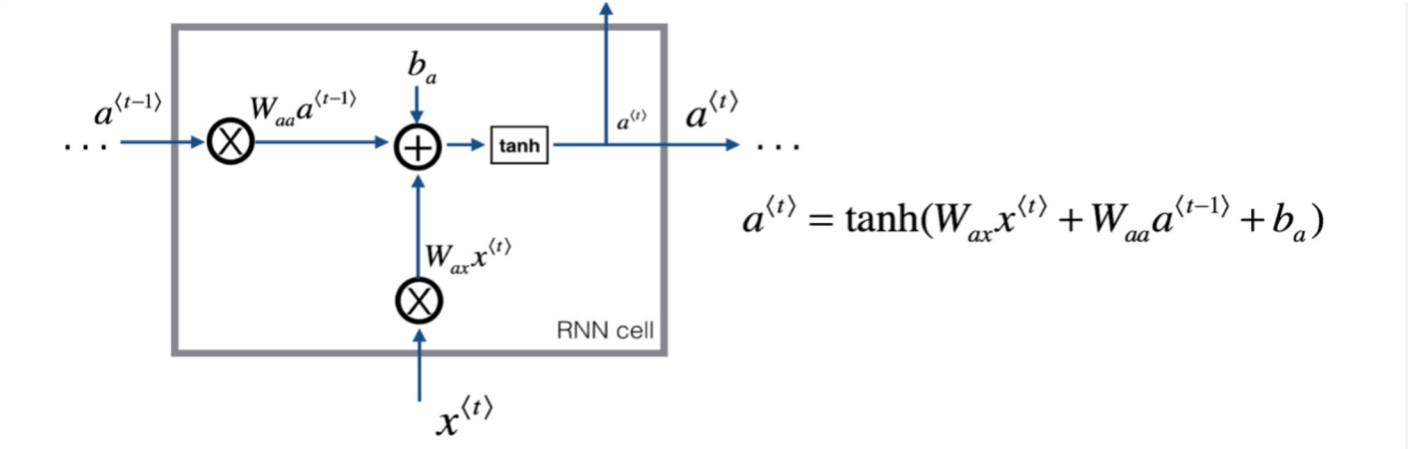


NOTE: RNN have vanishing gradient problem (backpropagation). LSTM and GRU are recurrent models alleviating this issue

Models

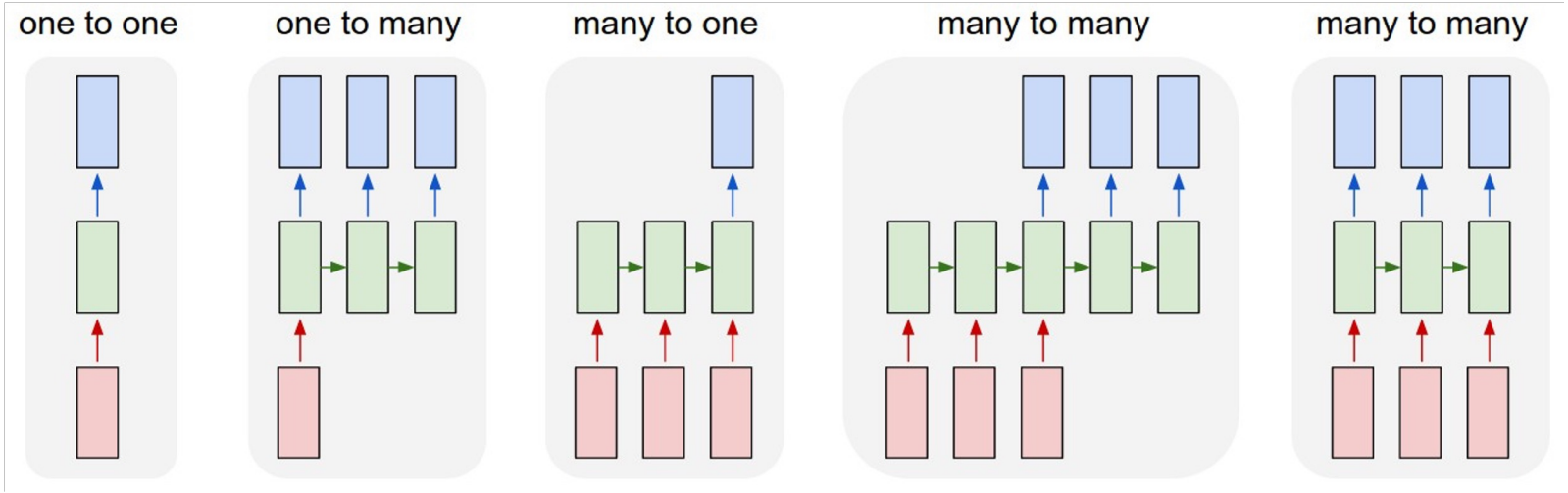
RNN

Inside a hidden state a^t



Models

RNN – A lot of flexibility



Sentence Classification

Translation

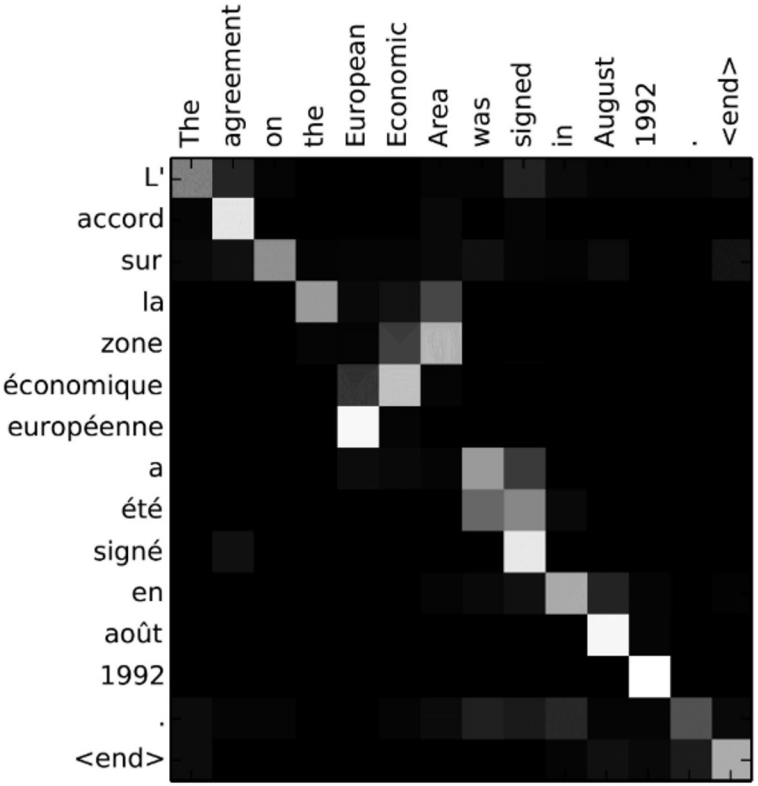
Models

Attention

Attention

Overview

- Helps modeling sequential data
- Allows concentration on a subset of attributes (weight)
- Weights are "attention"
- Eliminates the issue of distance or length of recurring networks
- Link with humans:
  Translation example ->
- Fun fact: from Montreal!



Attention

Formally

More specifically, in a traditional RNN encoder-decoder, for an input sentence represented by the sequence of vectors $x_1, x_2, ...x_T$, the encoder builds hidden states $h_1, h_2, ...h_T$ and the decoder uses the same context vector c for all predictions it makes. Usually, the context vector is the last hidden layer of the encoder (i.e. $c = h_T$). The addition of attention implies different context vectors c_i for each prediction. These context vectors are weighted sums over all the hidden states of the encoder.

28

Bahdanau_Listen_Before_You_Talk.pdf

Formally,

$$c_i = \sum_{i=1}^{T} \alpha_{ij} h_j$$

where $\alpha_{ij} = \text{softmax}[a(s_{i-1}, h_j)]_j$, s_{i-1} is the hidden state of the decoder producing word $i-1$ and a is a feedforward neural network jointly trained with all the other components of the system.

> Attention was initially integrated to existing models

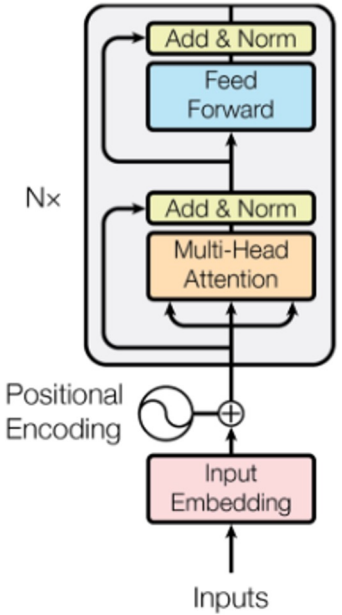
Models

Transformers

Transformers

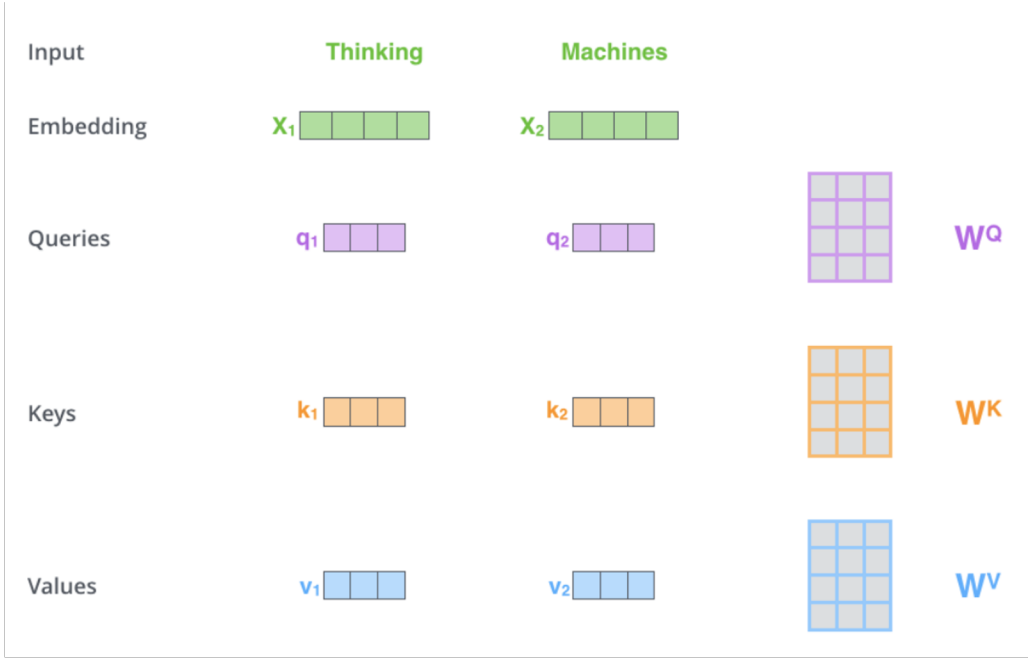
Overview

- Self-attention
  - Each « token » evaluated attention on each « token »
- Multi-head
  - Learning multiple representation sub-spaces
    - Ex with text: spelling, feelings, accent, …
- Useful for sequence-to-sequence (but not sequential)
- Encoder - Decoder



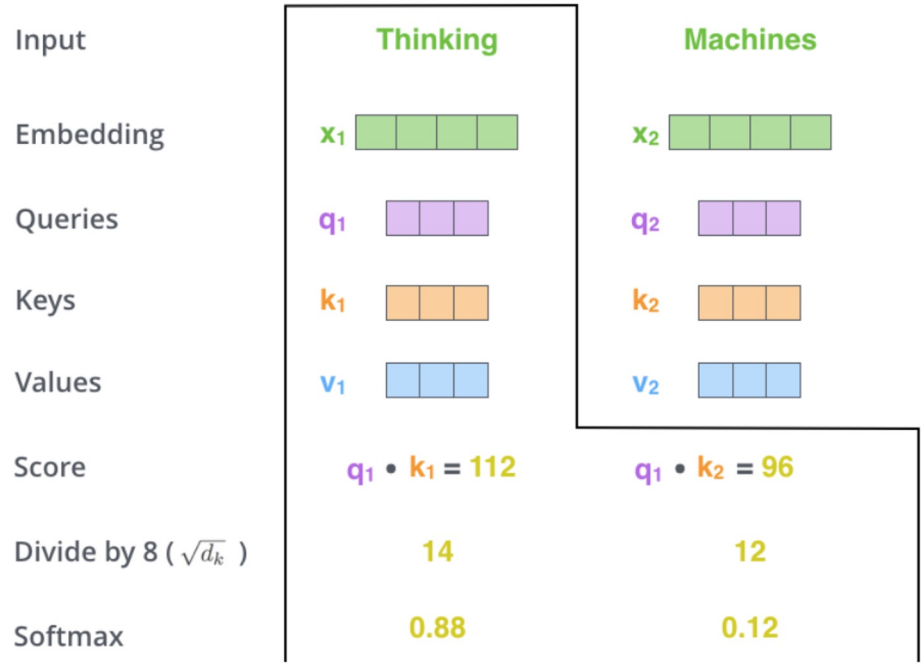
Transformers

Self-attention calculation – Parameters (W)



Transformers

Self-attention evaluation



Note how this is not sequential

Models

Bert

BERT

Bidirectional **E**ncoder **R**epresentations from **T**ransformers

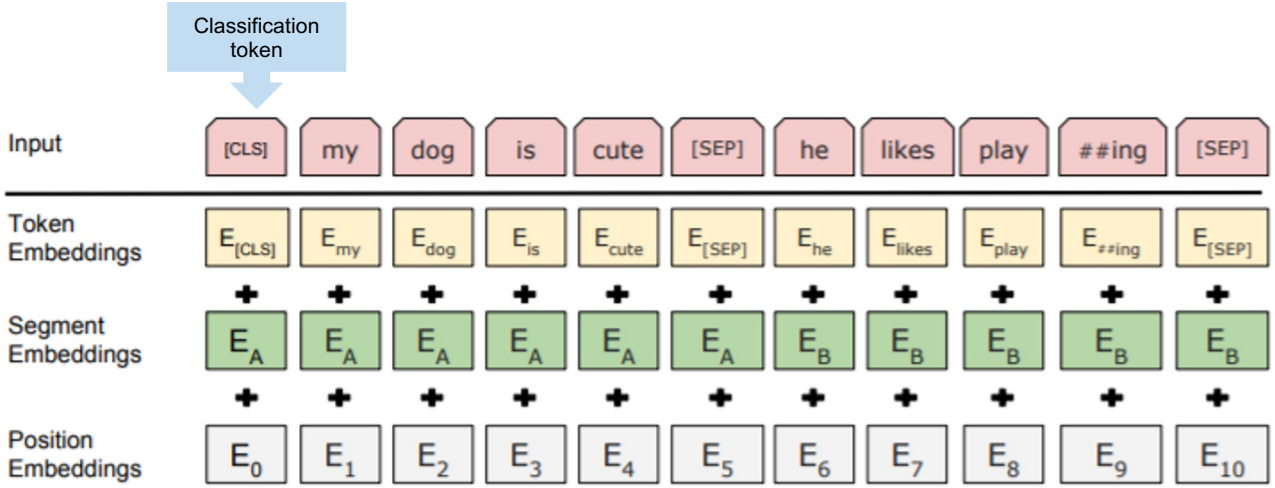
- Uses transformers (and more)
- Pre-trained on natural language
- Goal: Transfer learning

BERT

Becoming sequential

> Transformers are not sequential (self-attention), language is!

> Solution: Positional encoding

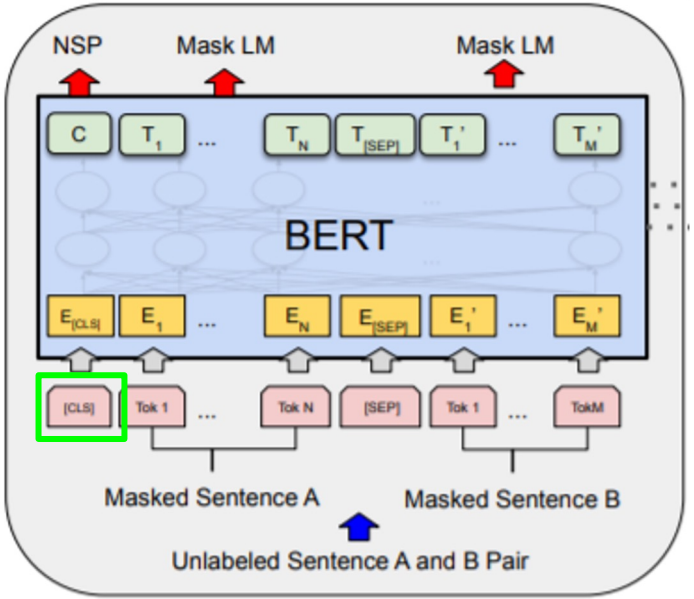


BERT

Unsupervised pre-training

1. Mask Language Model
   - Hide words in input and learns to predict them at output
2. Next Sentence Prediction:
   - Learns to classify if 2 sentences follow one another in a document

> After, train on your specific task (fine-tuning), hence *transfer learning.*

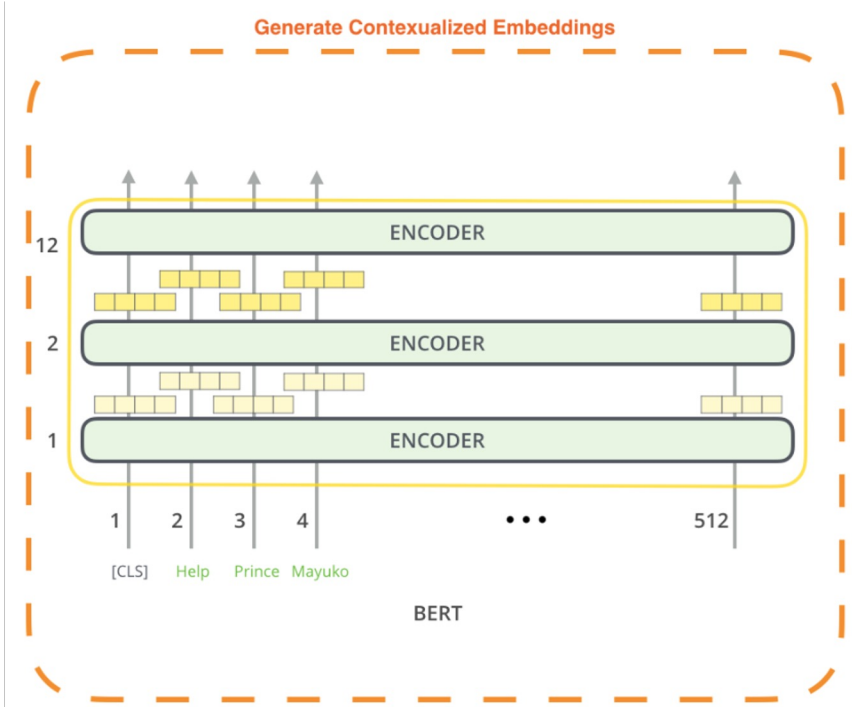


Pre-training

BERT

Magnitude – BERT Base

- 12 layers of transformers
- 12 self-attention heads
- 512 tokens
- Tokens ~ number of words
- Super parallel -> GPU
- $O(n^2)$, where n is amount of tokens



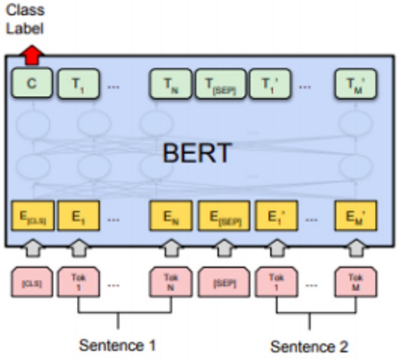
BERT

Bert – ~~Word2Vec~~ Context2Vec

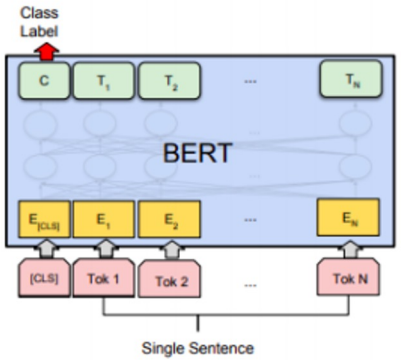
- Context-free models such as word2vec or GloVe generate a single word embedding representation for each word in the vocabulary, where BERT takes into account the context for each occurrence of a given word.
- For instance, whereas the vector for "running" will have the same word2vec vector representation for both of its occurrences in the sentences "He is running a company" and "He is running a marathon", BERT will provide a contextualized embedding that will be different according to the sentence.
- Sentence embedding: through [CLS] token or average of last hidden layer output

BERT

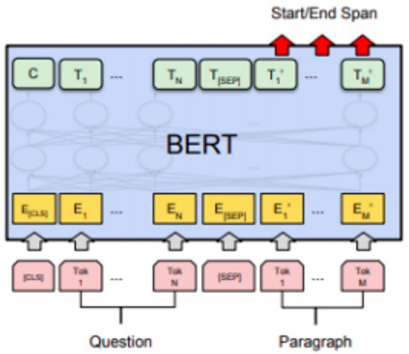
Tasks



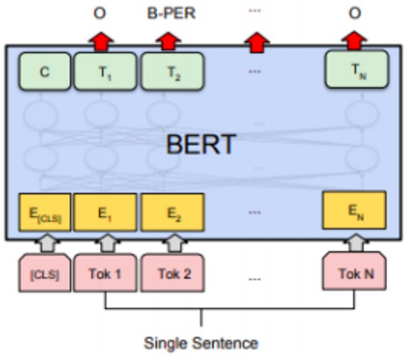
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA



(c) Question Answering Tasks: SQuAD v1.1



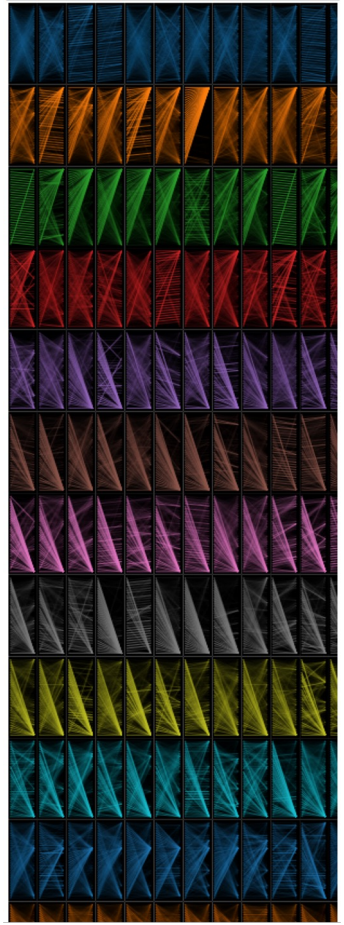
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BERT

Explainability - Through attention

- On all the model
  - 12 layers (horizontal)
  - 12 heads (vertical)
- Different patterns
- Can come from pre-training
- Specific to the trained model and input text
- Here: same template as Enbridge split text

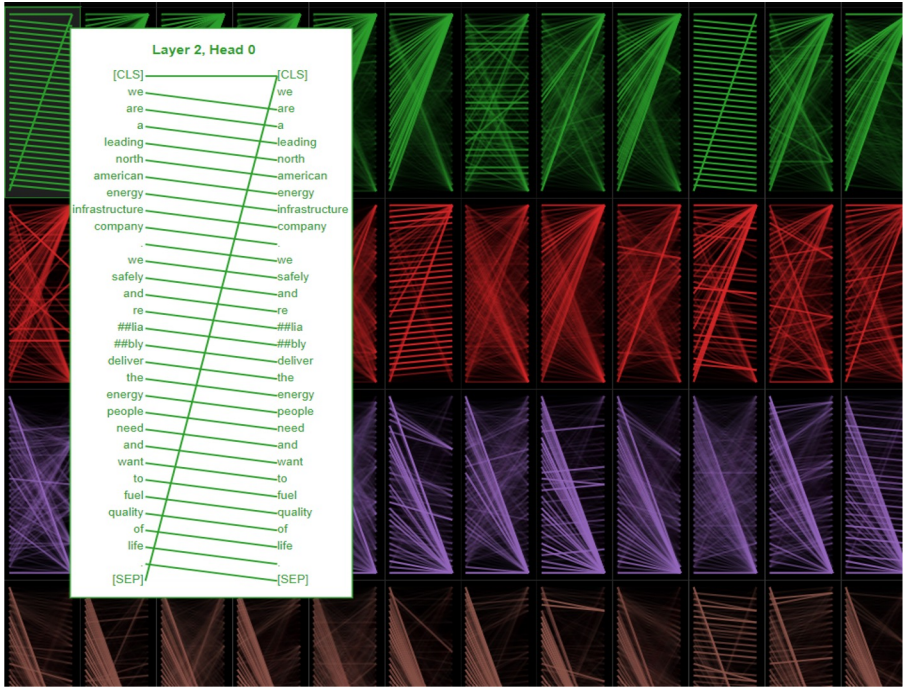
“We are a leading North American energy infrastructure company. We safely and reliably deliver the energy people need and want to fuel quality of life.”

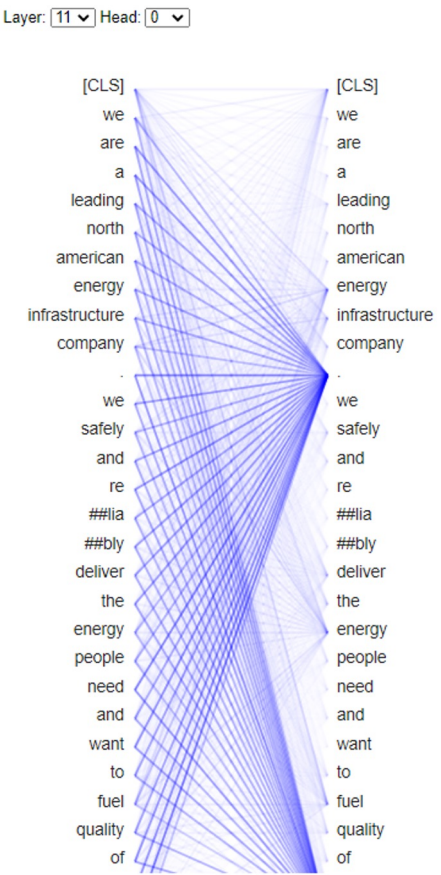


BERT

Explainability - Through attention

- On a particular layer and head
  - 12 layers
  - 12 heads



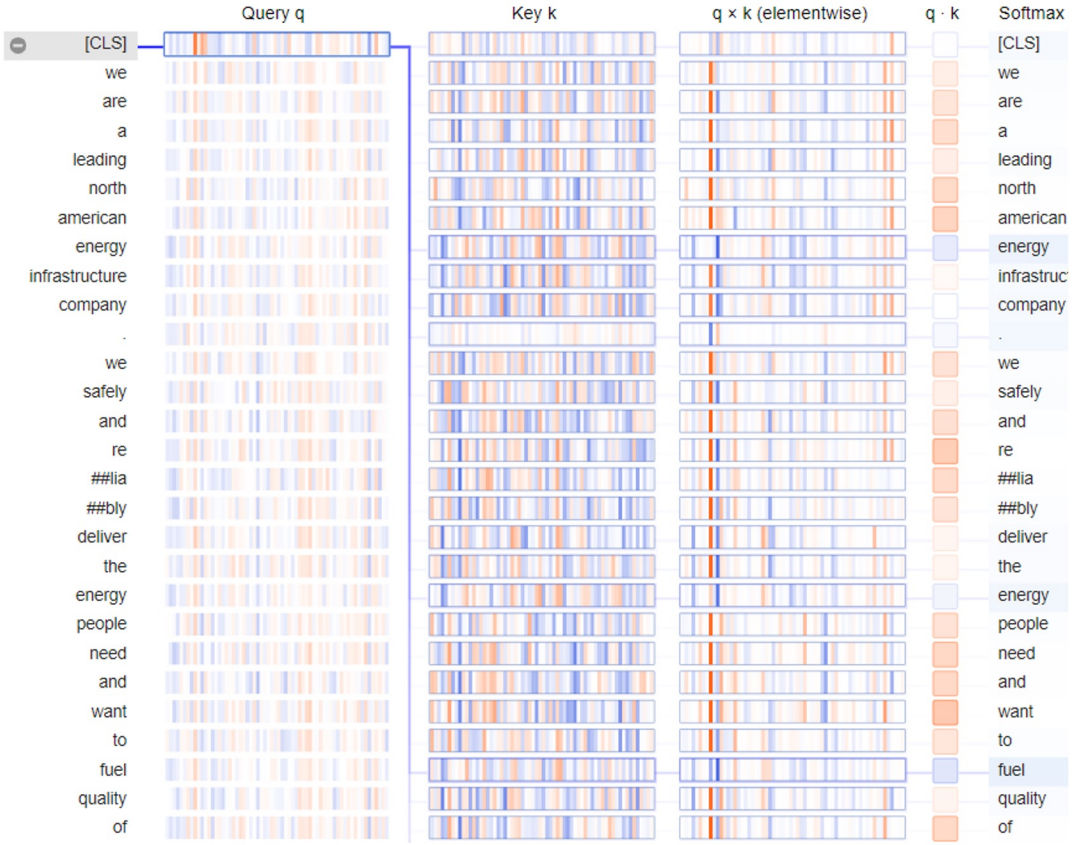


BERT

Explainability - Through

> On a token…
  - …in one layer
  - …in one head

> Here:
  - Last layer
  - CLS token
  - Attention is on « energy » and« fuel »



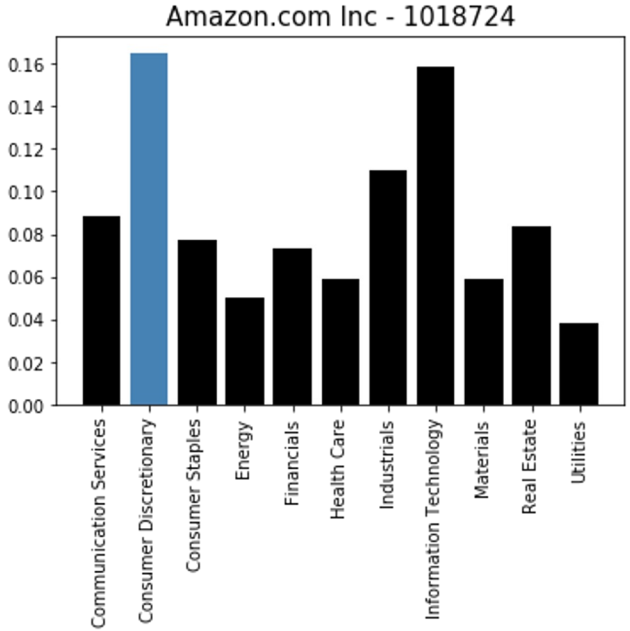
NLP in finance

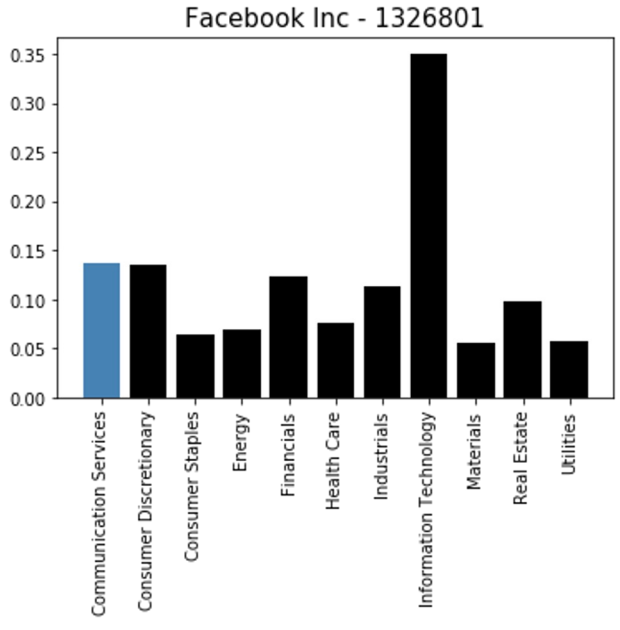
Examples

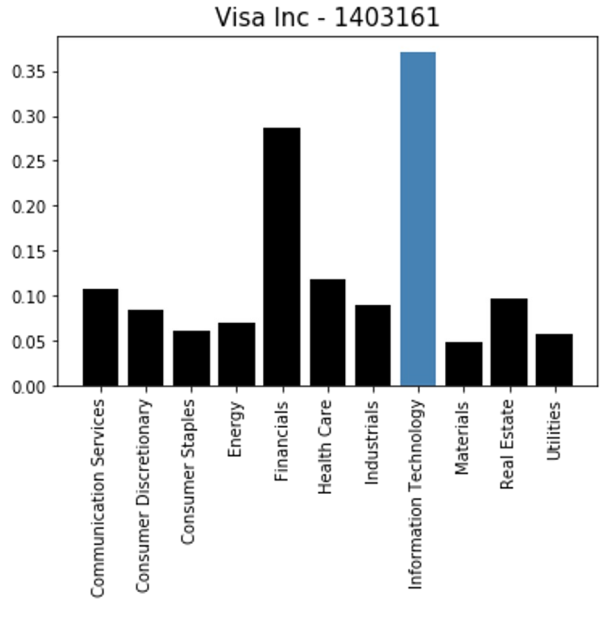
NLP in Finance

Neural Networks Method

Bert at CDPQ - Soft Predictions Reveal Underlying Sectors







NLP in Finance

Sentiment Analysis – Variations in 10k - 10Q

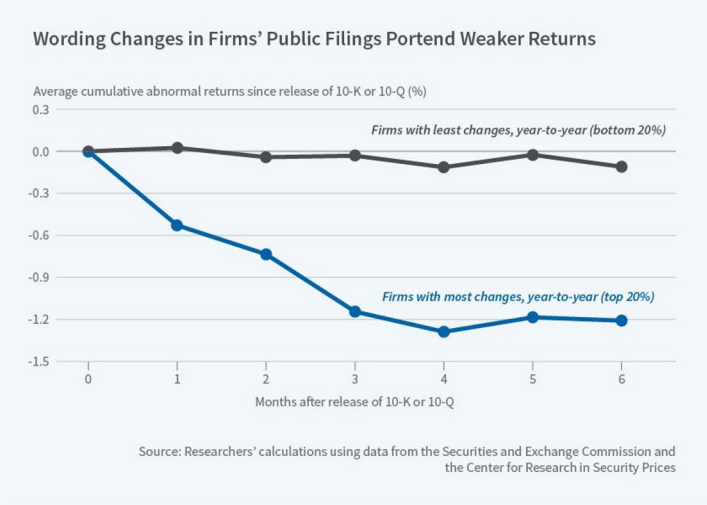
Rule Based Method

> Change impacts return

- Long "non-changers" and short "changers" earned a statistically significant value-weighted abnormal return of between 34 and 58 basis points per month—between 4 and 7 percent per year—over the following year

> Metrics of change

- Term frequency
- Number of operations needed to go from one document to the other
- Document comparison software (Textual Similarity Measure)



NIVELOP

NLP in Finance

Sentiment Analysis – Positivism

Rule Based Method

> How?
  - Count proportion of negative words
  - Correlate with future returns

> Loughran and McDonald (2011) dictionary
  - Financial dictionary
  - 3 lists of sentiment word
    – master word list (80,000+)
    – positive (350+)
      ▪ able, abundance, acclaimed, accomplish
    – negative (2300+)
      ▪ abandon, abdicate, aberrant, abetting

For an example, see S&P Quantamental NLP Research paper at https://www.spglobal.com/marketintelligence/en/documents/mi-research-qr-nlp-part-ii-180912-new.pdf

- Q&A section driving more (prepared remarks seem to dilute)
- Discordance between Prepared remarks and Q&A not driving outperformance empirically
- CEO's sentiment more predictive

NLP in Finance

Sentiment Analysis – Behavioral

Rule Based Method

> How?
  - Evaluate text with rules
  - Correlate with future returns

> Gunning fog index
  - Measures the number of years of formal education needed to understand (e.g.16 is undergraduate)
  - Input:
    – average number of words per sentence
    – proportion of polysyllabic words (threshold: 3 syllables or +)

> Quantity of numbers in the text

See S&P Quantamental NLP Research paper at https://www.spglobal.com/marketintelligence/en/documents/mi-research-qr-nlp-part-ii-180912-new.pdf

- Basic language outperforms more complex one
- Sentiment and behavioral signals uncorrelated.
- Sentiment (previous slide) and behavioral based signal has *additive* predicting power above the 8 commonly used alpha and risk signals (Beta • Market capitalization • Valuation • Price momentum • Asset growth • Gross Profitability • Analyst Revision • Earnings Surprise)

NLP in Finance

Sentiment Analysis – Key Phrases Detection

Statistical Method

> What?

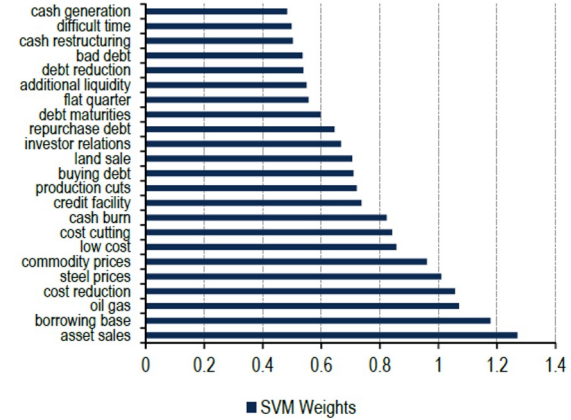
- Identify sequence of word predicting an outcome
  - Ex: 2 words sequence mentioned in earning calls leading to debt default in next 12 months*

> How?

- Create n-grams from text
- Get tf-idf for each n-grams
- Use an interpretable model (SVM, Tree,…)
- Input n-gram/tf-idf representation of document and its target (outcome)
- Select n-grams with highest feature importance

Figure 7: Examples of key phrases from earnings calls associated with High Yield defaults

The SVM weights are the estimated coefficients as a result of training the Support Vector Machine model as described in the methodology section. The higher the weight indicate a higher importance for that phrase. We only show a sample of thousands of phrases to illustrate the mechanics of the model.



** Doc shared by Ignacio in email*

NIVELOP

NLP in Finance

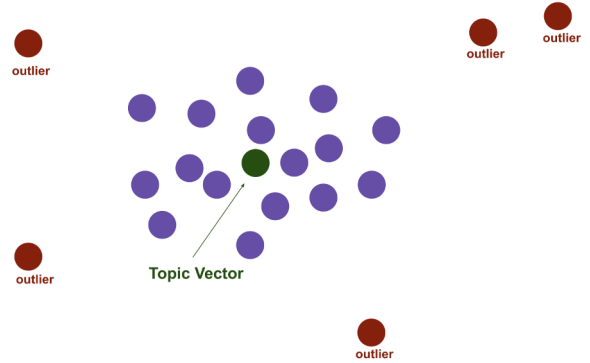
Topic Modeling

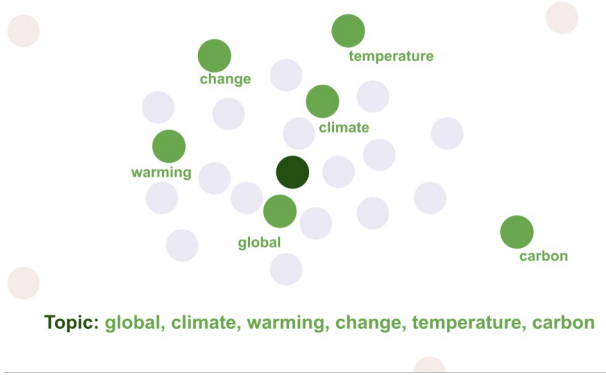
> What are these documents talking about?

> No categories or label (unsupervised)

- LDA[1] and NMF
  - Good starting points, but hyperparameters tuning complicated
- Topic as vectors[2] (2 techniques)
  - Embed documents (and words)
  - Cluster documents
  - Words as centroids (tf-idf on clusters of documents)
  - Topics are words close to centroid (topics are words with highest tf-idf score for clustered doc compare to rest of documents)

Statistical and Neural Network Method





[1] https://towardsdatascience.com/your-guide-to-natural-language-processing-nlp-48ea2511f6e1

[2] https://arxiv.org/abs/2008.09470

NIVELOP

Annex

Annex

Accuracy – 10K Predicting GICS Sectors (512 token)

	Bag of Words	tf-idf	Bert
Naïve Bayes	85	59	
SVM	86	87	
FeedForward Network	93	92	
Bert			94

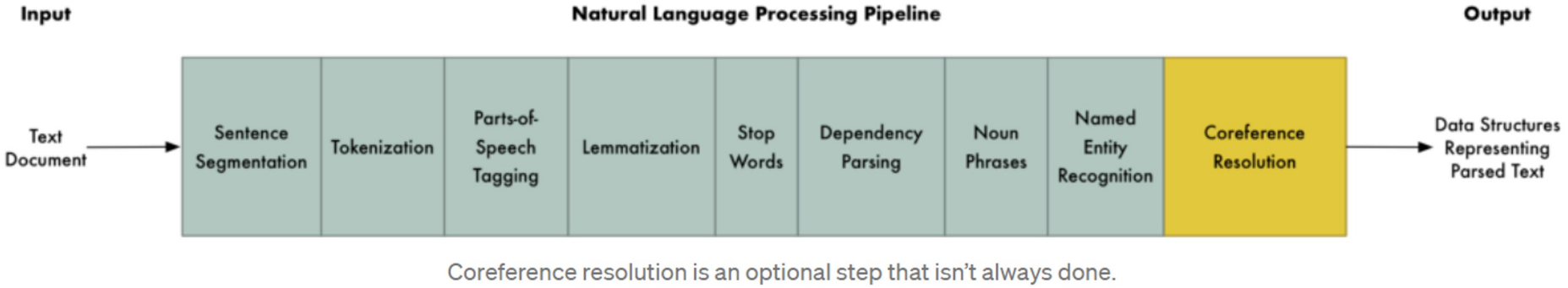
Don’t try to kill a fly with a tank

Bert is expensive, even once trained.

But, don’t try to summarize or translate with tf-idf.

Annex

Example of NLP Pipeline



Coreference resolution is an optional step that isn't always done.