Natural Language Processing and Learning

Andrea Galassi

12 April 2019

Intelligent Systems M

Natural Language Processing

Cross-field discipline: mixture of linguistics, artificial intelligence, …

Aim: analyze, represent, understand, and re-create natural language

This can be applied both to speech and to textual documents

Summary

1) NLP tasks
2) How to address these tasks
   - NLP features
   - Basic architectures for NLP
3) Case Study: Argumentation Mining
4) Conclusion

NLP Tasks: document-level analysis

- Document Classification
  - News: sport, politics, …
  - Books: thriller, sci-fi, fantasy, …
  - Website: scientific, news, opinion, …
- Sentiment Analysis
  - Reviews: positive or negative
  - Tweets
  - Essays: for or against something
- Abusive content detection
  - Racism
  - Bullying
- Fake News detection

NLP Tasks: word/sentence-level analysis

- Named entity recognition
  - Persons, locations, organization
- Structure extraction
  - Chapter/topics separation
  - Argument Mining: discussion in the text
- Part Of Speech (POS) tagging
  - Grammatical role of words

NLP Tasks: generation

- Question Answering
  - Questions over a specific document
  - General knowledge questions
- Image Captioning

"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

- Translation
- Summarization
- Conversational AI
  - Chat-bots
  - Digital-assistants (Alexa, Siri, Cortana)

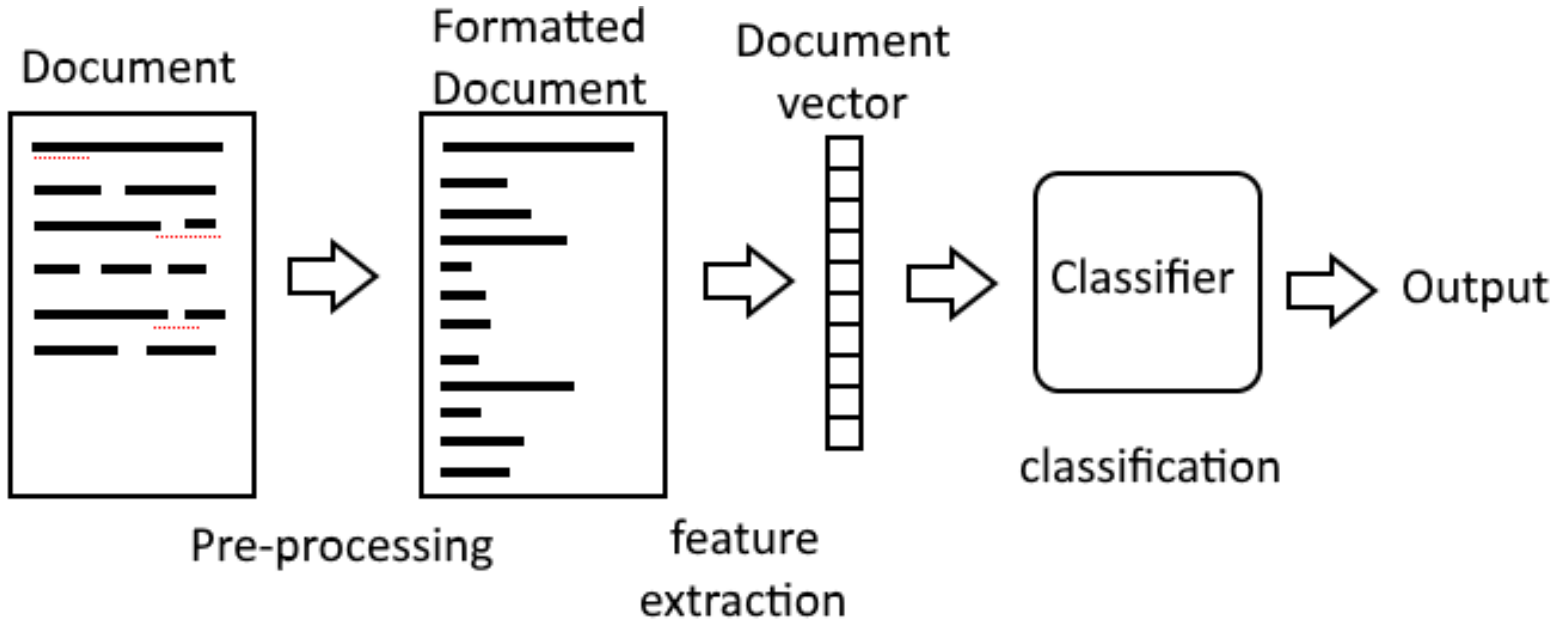
Supervised Learning setting

Given

- a collection of documents (Corpus)
- the desired outputs for each document

Train an automatic system to perform a specific task on new documents

Document-Level Analysis Pipeline



Pre-processing: correction of grammar mistakes, substitution of symbols, ...

Document vector: document represented as a vector of numerical features

Document Features: Bag of Words

In the corpus (collection of documents) there are n different words

1 document = 1 n-dimensional vector

Each word is a dimension

The value along that dimension is the number of times that the word occurs in the document: term frequency (TF)

Document D: "I like football, you like basketball"

Word W	I	You	Like	Love	Football	Basketball
TF(W,D)	1	1	2	0	1	1

Document Features: Bag of Words

Some words are very common and not significant: prepositions, pronouns, some verbs.

How to exclude irrelevant words?

- Stop-words: do not consider most common words
- TF-IDF = Term Frequency * Inverse Document Frequency

IDF(W) = log (# documents / # documents containing W)

Document Features: n-grams and skip-grams

Similar to BOW

- N-grams: Vocabulary made of sequences of n adjacent words
- Skip n-grams: Vocabulary made of sequences of n words within some distance range

Document D: «Sport is very important»

1-grams: «Sport», «is», «very», «important»

2-grams: «Sport is», «is very», «very important»

Skip 2-grams: «Sport very», «sport important», «is important»

Document Features: Statistics

- Number of words
- Average length of a sentence
- Number of sentences
- Average length of words (in characters)

How can these be helpful???

Examples: classification for the genre of the document, quality of an essay, sentiment (?)

Document Features: presence of Cue Words

Presence or lack of specific words

- Sentiment words: «good», «bad», «awful», «excellent», «lovely»
- Topic-related words: «touchdown», «kick-off», «election», «parliament»
- Curses or offensive words: «f**k», «c**t», «f****t», «p***y»

Document Features: considerations

The n-grams features do not take into account semantic similarity

"Football" is equally different from "Basketball" and "Potato"

Cue Words take into account semantics, but are specific to the domain and the task

To have more semantic information it is necessary to work at word-level

Document Features: sequence of words

Document = sequence of words

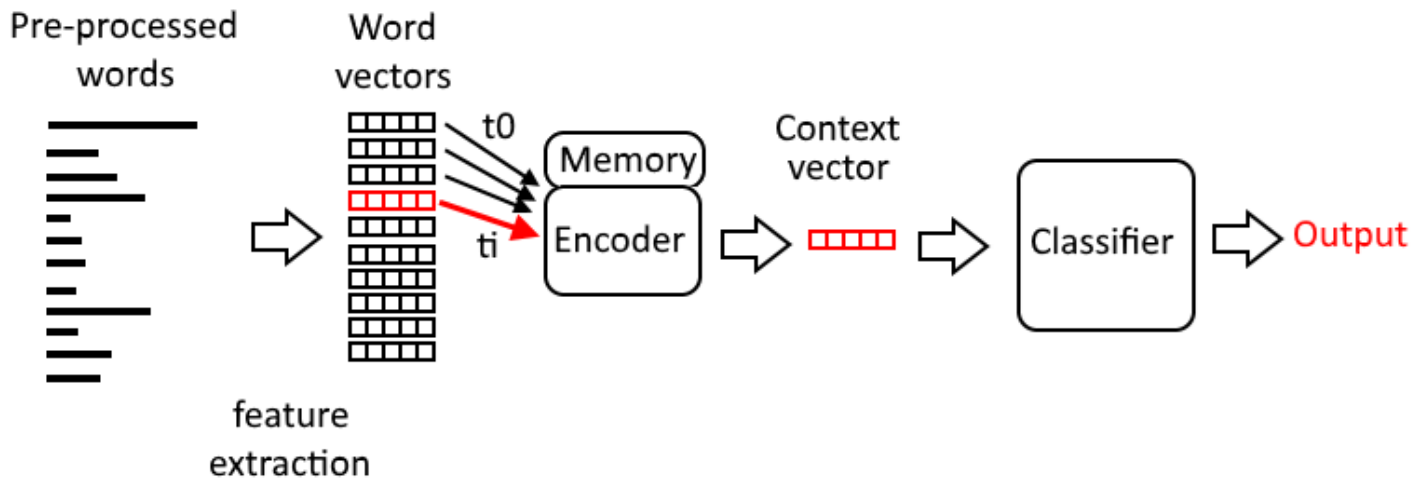
If the word can be represented as a feature vector, a document is simply a sequence of feature vectors

Word-Level Analysis Pipeline

The words must not be processed independently, because the context is important

At each time step, a word vector is fed to the decoder and it's classified

The encoder has a memory of the words it has seen before, and produces the context vector accordingly



Word Features: exploiting external knowledge

If a knowledge base is available, it is possible to lookup for information regarding the word

Examples of knowledge bases:

- Databases
- Ontologies
- Online resources

Word Features: word embeddings

Using machine learning techniques, each word is associated with a high dimensional vector

It is possible to train a model over a specific corpus or to use the model pre-trained over other very large corpora

The training is UNSUPERVISED

Similar words are mapped to nearby region of the vector space

Examples: word2vec, GloVe, ELMO, fast text, …

From word embeddings to document feature

Document-level and word-level features are not mutually exclusive

It is possible to process word-level features to obtain a representation of the document

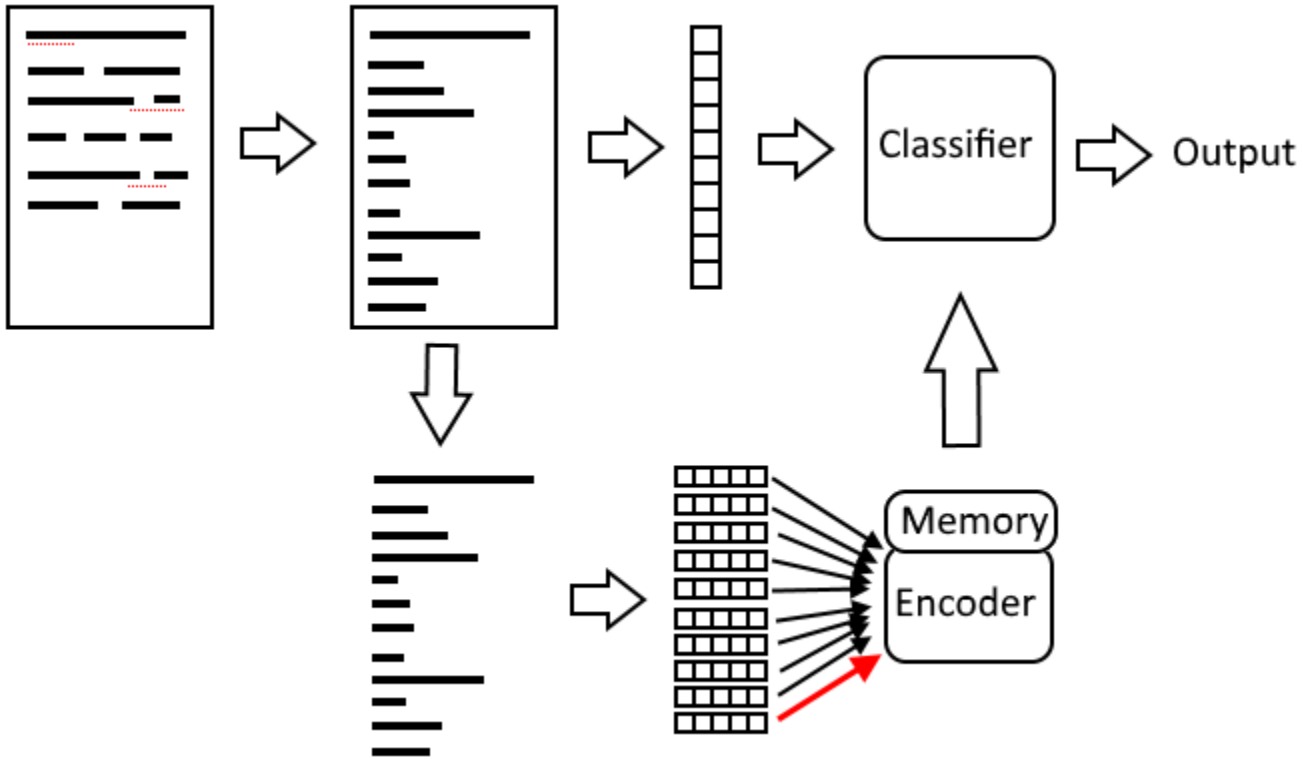
Most obvious way: average word embedding

Problems:

- Lot of useless words that introduce noise
- There is no concept of sequence: the same words in different order would produce the same result

Document Features: Encoder final state

The final state of the memory of the encoder captures the information about the whole text. It can be used as document-level feature



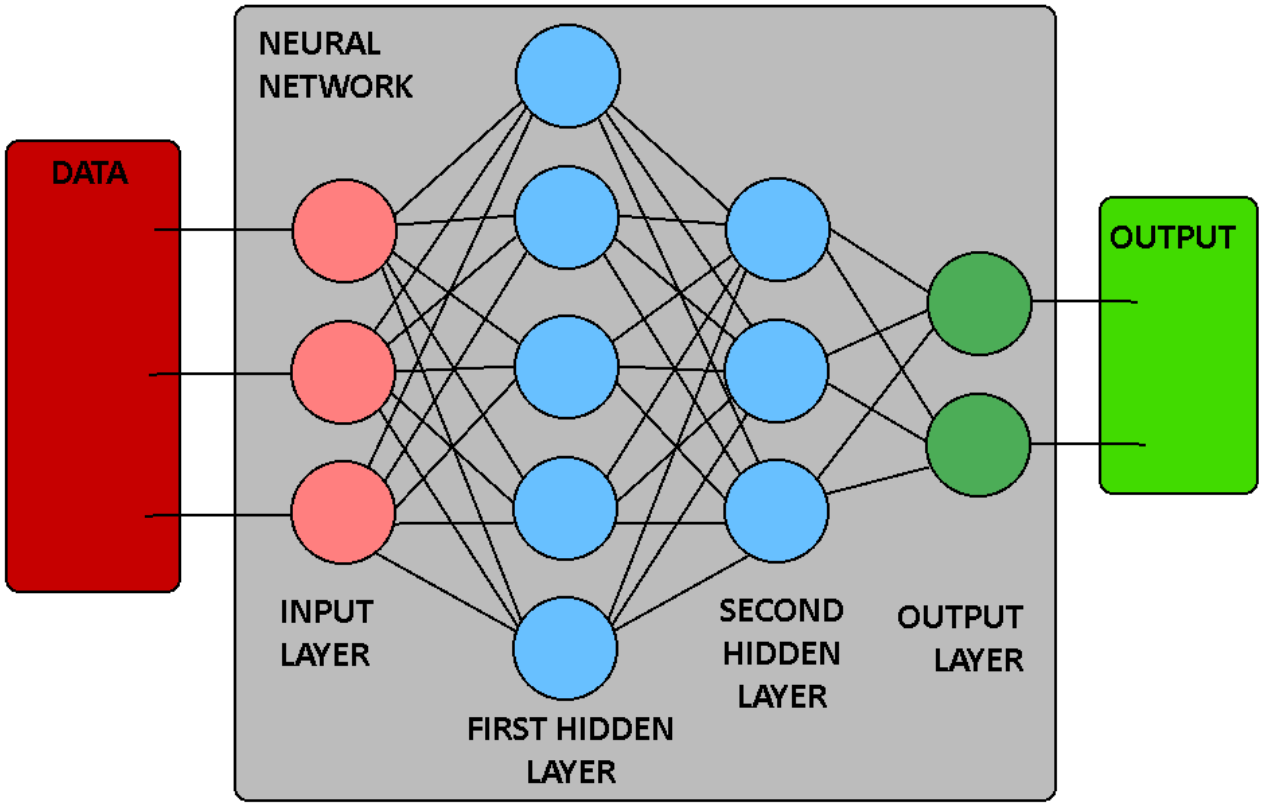
Neural Networks in a nutshell

Each layer receive a vector as input.

Each neuron computes a weighted sum of all its inputs and then applies a non-linear function.

The outputs of all the neurons of a layer are the input to the next layer.

The system learns the weights of the connections

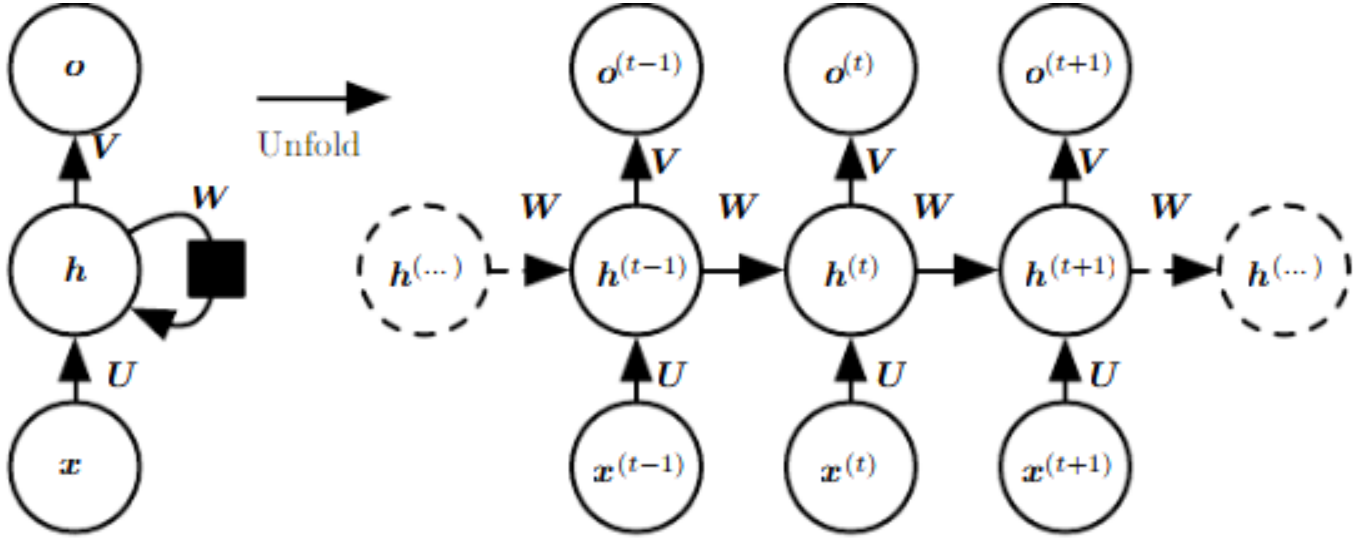


Encoder Model: Recurrent Neural Networks

The encoder can be built through recurrent neural networks

They weights matrices U, W, and V are learned altogether

At each time-step, they take a word **x** as input, compute a new state **h** (the memory), an produce an output **o**

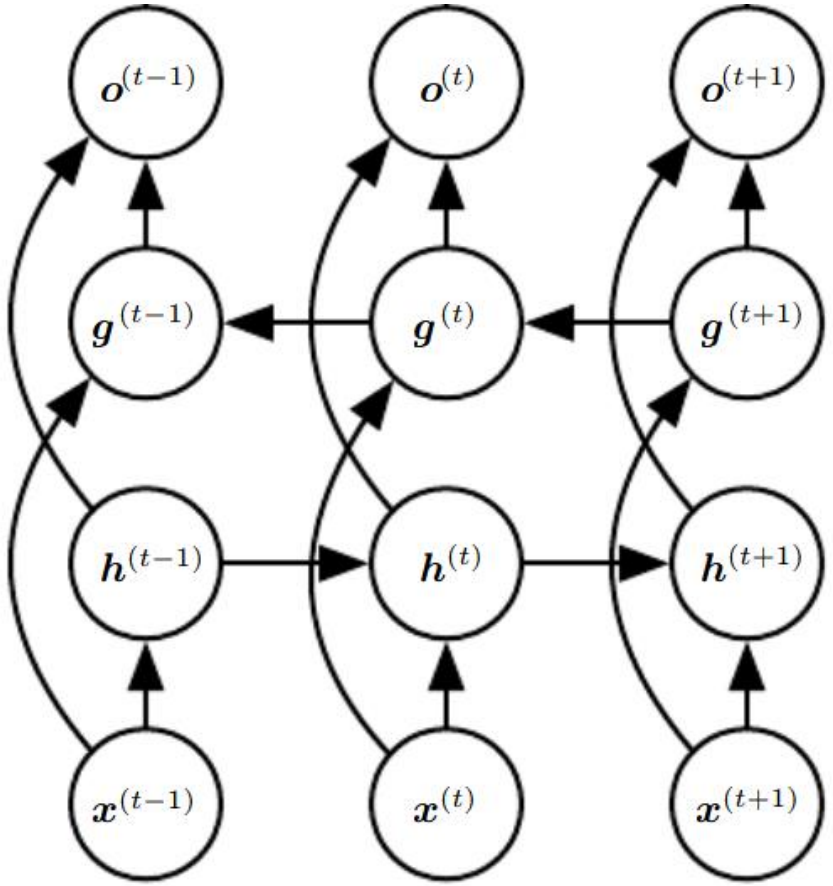


Deep learning (Goodfellow, Bengio, Courville, 2016)

Encoder Model: Bidirectional RNNs

Words in a document do not only depend on previous words, but also on following ones.

Bidirectional RNNs maintain two hidden states: one (**h**) depends on the previous words, one (**g**) on the following ones



Deep learning (Goodfellow, Bengio, Courville, 2016)

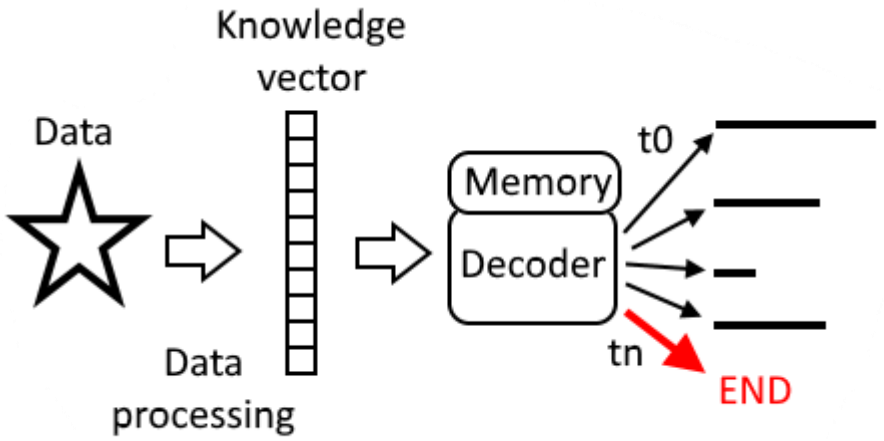
Generation Pipeline

The knowledge vector is fed to the decoder through multiple interaction.

The decoder produces a word at each time step

The decoder must have a memory to remember what it has generated

=> It can be a RNN!



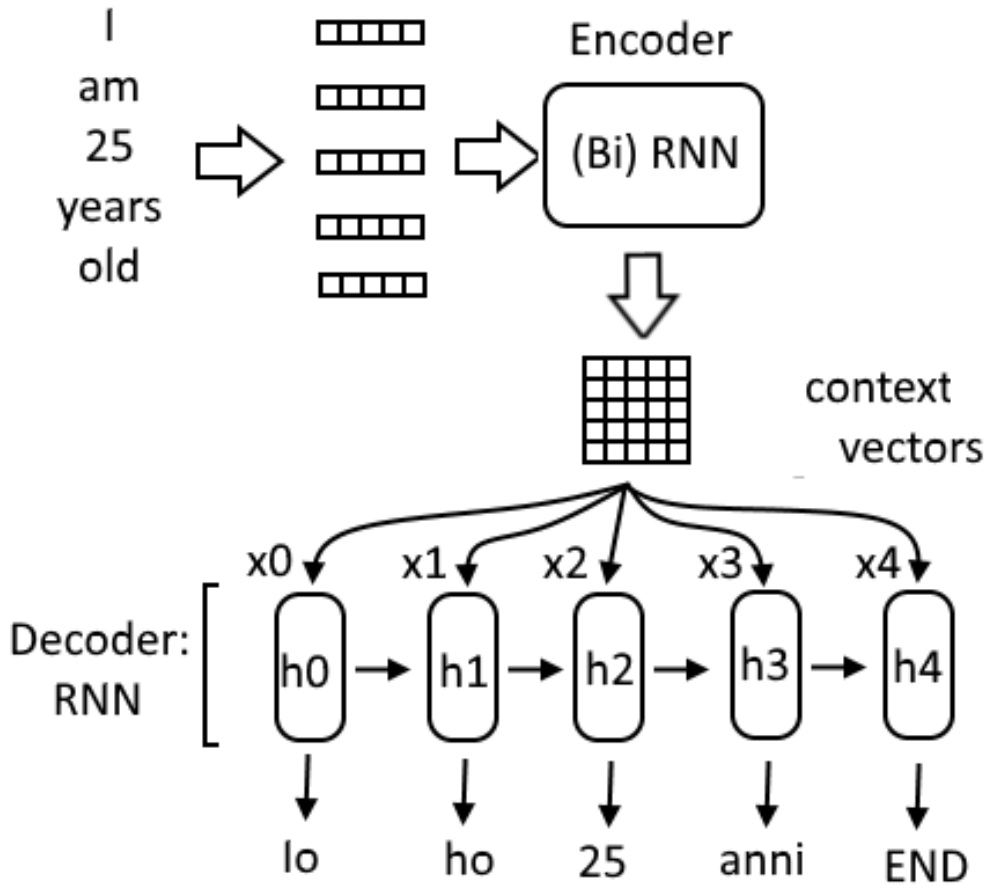
Example: Machine Translation

Encoder-Decoder architecture

The encoder is a (bi)RNN

The decoder is a RNN

At each time step, the decoder is fed with the whole output of the encoder



Neural Attention

Are all the words equally important for each instance of each task?

It would be helpful to have a mechanism to discriminate between relevant and irrelevant words, given the present focus

The idea behind the attention mechanism is to give a relevance weight to each input, so to filter out irrelevant words

Other positive effect: interpretable result! The higher the weight, the more important is the input

Neural Attention

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

Show, Attend and Tell: Neural Image Caption Generation with Visual Attention (Xu et al., 2015)

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Deriving Machine Attention from Human Rationales (Bao et al., 2018)

Task: Hotel location

you get what you pay for . not the cleanest rooms but bed was clean and so was bathroom . bring your own towels though as very thin . service was excellent , let us book in at 8:30am ! **for location and price , this ca n't be beaten** , but it is cheap for a reason . if you come expecting the hilton , then book the hilton ! for uk travellers , think of a blackpool b&b.

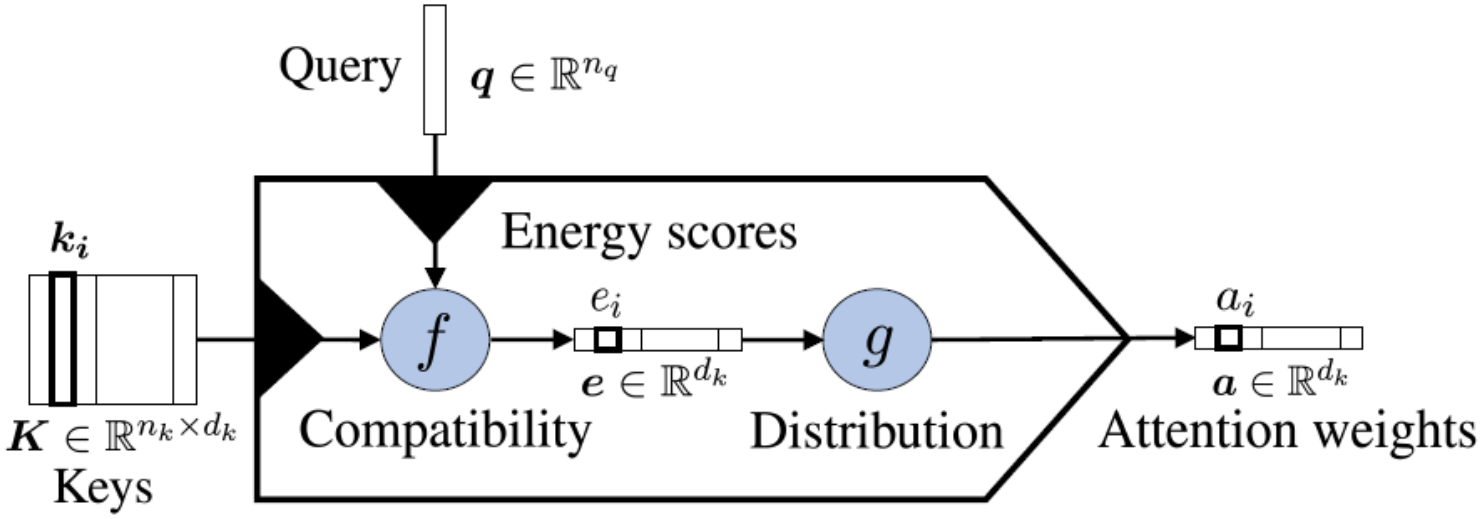
Task: Hotel cleanliness

you get what you pay for . **not the cleanest rooms but bed was clean and so was bathroom** . bring your own towels though as very thin . service was excellent , let us book in at 8:30am ! for location and price , this ca n't be beaten , but it is cheap for a reason . if you come expecting the hilton , then book the hilton ! for uk travellers , think of a blackpool b&b.

Task: Hotel service

you get what you pay for . not the cleanest rooms but bed was clean and so was bathroom . bring your own towels though as very thin . **service was excellent** , let us book in at 8:30am ! for location and price , this ca n't be beaten , but it is cheap for a reason . if you come expecting the hilton , then book the hilton ! for uk travellers , think of a blackpool b&b.

Neural Attention



Attention, please!
A Critical Review of Neural Attention Models in Natural Language Processing
(Galassi, Lippi, Torroni, 2019)

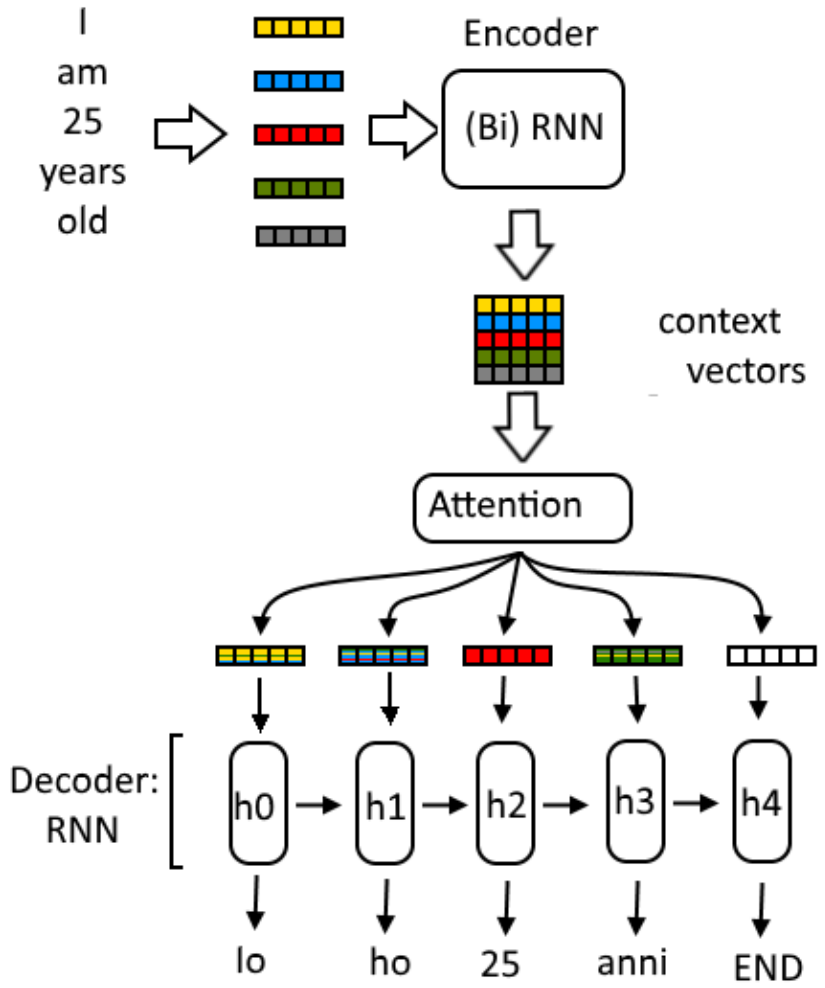
Keys: inputs (word vectors)

Query: possible additional information

With the attention weights, it is possible to compute a weighted sum of the keys, and obtain a document-level feature without considering noisy information

Using attention for Machine Translation

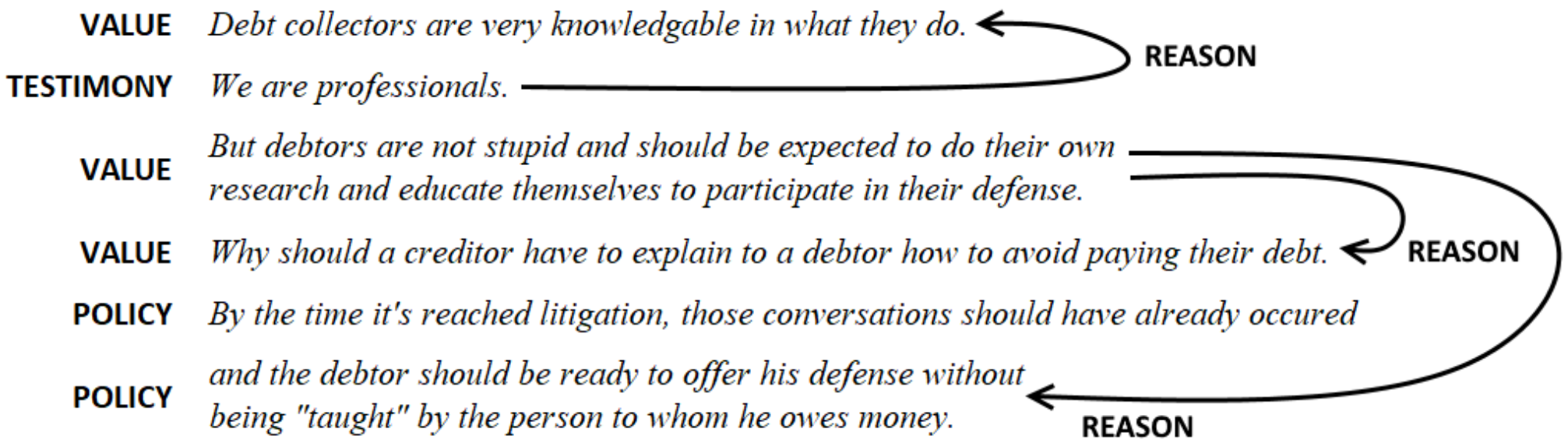
Now, at each time step, the decoder receives an input focused according to the new word that must be generated



Argumentation Mining

Aim: reconstruct the argumentation inside a document

Use: create structured knowledge out of documents



Argument mining with structured SVMs and RNNs (Niculae, Park, Cardie, 2017)

Argumentation Mining

5 steps:

1) Divide the document into components
2) Identify the argumentative components
3) Classify the components (e.g. evidences and claims)
4) Find which relations do exists between the components
5) Classify the relations (e.g. support and attacks)

They can be performed in a pipeline, or at the same time (end-to-end)

Argumentation Mining: examples

End-to-end AM as sequence tagging

Using a RNN, each word is tagged with 4 labels:

- BIO encoding: O = not part of an argument, B = begins an argument, I = continues an argument
- Type of the component
- Argumentative distance between this component and the one it relates to
- Type of the relation

Since	it	killed	many	marine	lives	,	tourism
(O,⊥,⊥,⊥)	(B,P,1,Supp)	(I,P,1,Supp)	(I,P,1,Supp)	(I,P,1,Supp)	(I,P,1,Supp)	(O,⊥,⊥,⊥)	(B,C,⊥,For)
has	threatened	nature	.				
(I,C,⊥,For)	(I,C,⊥,For)	(I,C,⊥,For)	(O,⊥, ⊥, ⊥)				

Neural End-to-End Learning for Computational Argumentation Mining (Eger et al., 2017)

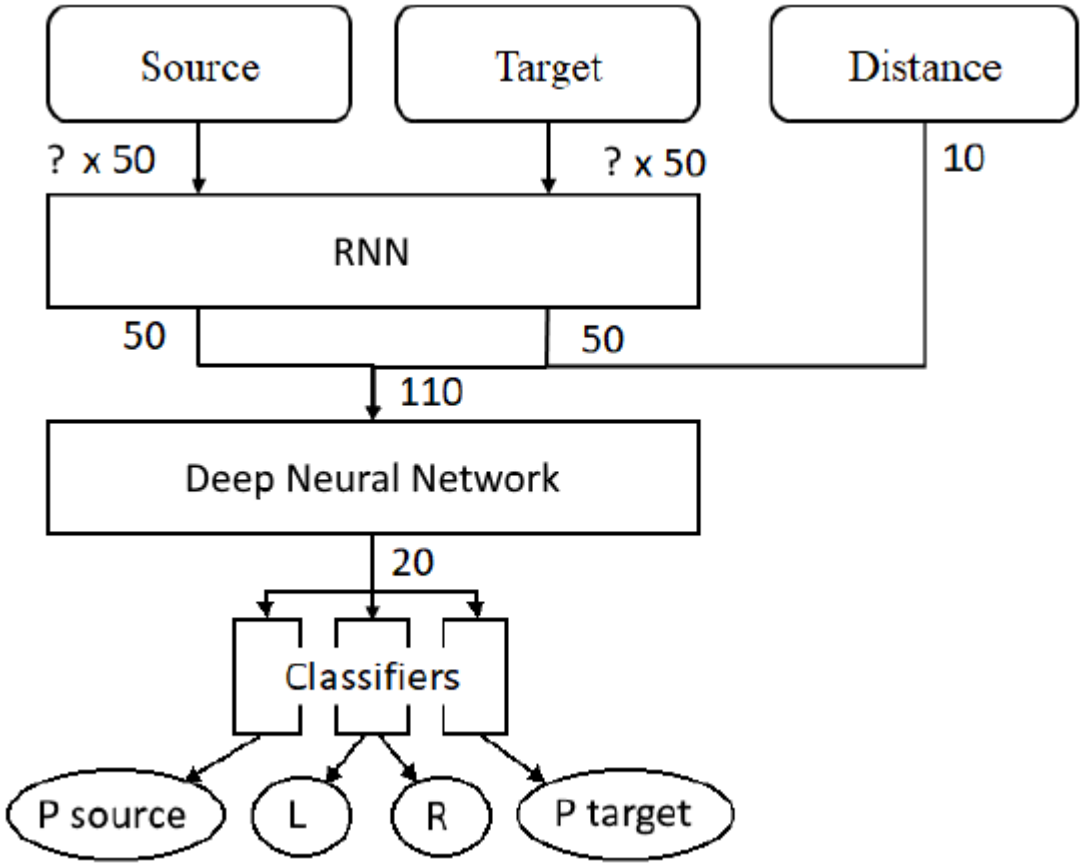
Argumentation Mining: examples

Multi-task AM

Input: couples of propositions of the same document (source, target)

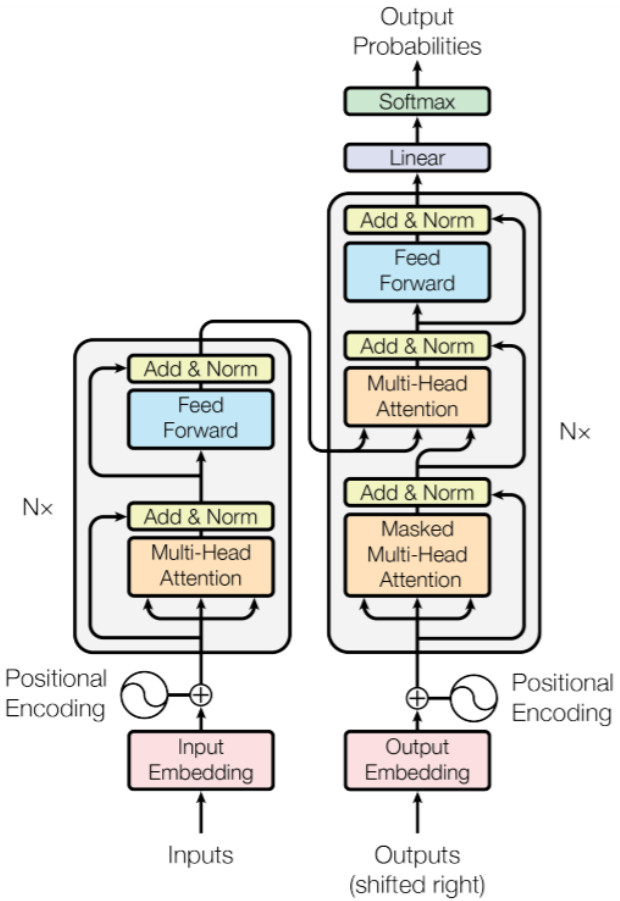
The RNN is used as embedder

3 parallel classifiers assign the labels: type of the source, type of the target, if a relation does exists (L), and its type (R)

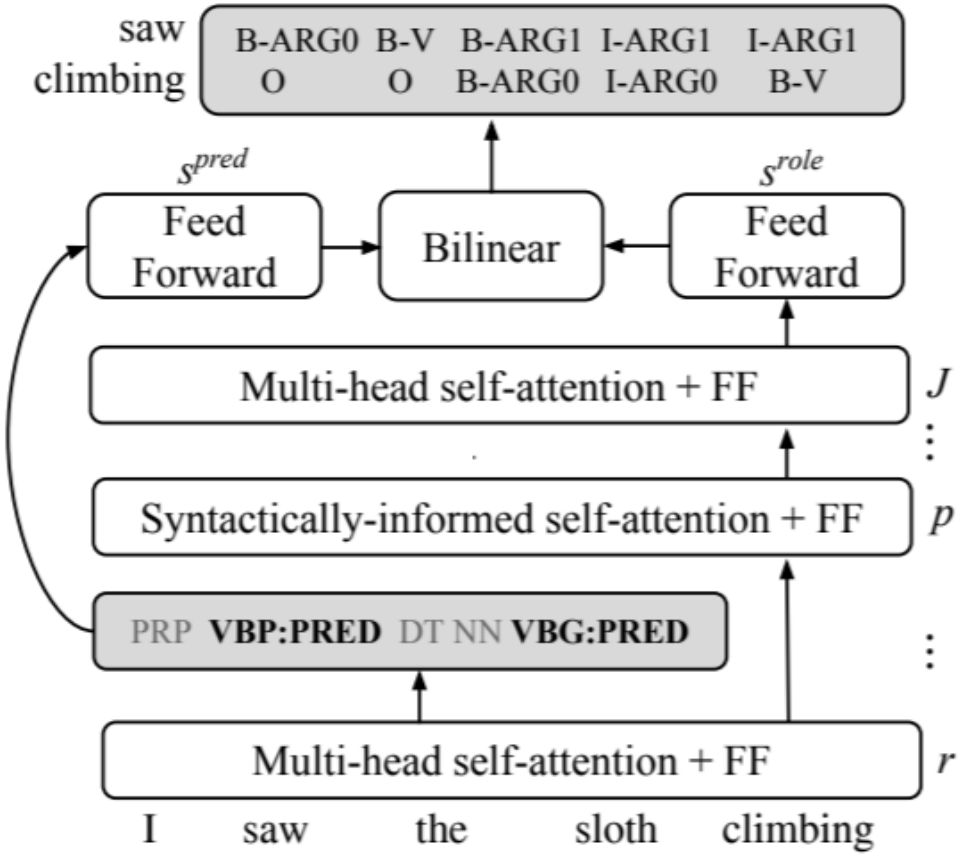


Argumentative Link Prediction using Residual Networks and Multi-Objective Learning
(Galassi, Lippi, Torroni, 2018)

State of the Art architectures are way more sophisticated



Attention is All You Need (Vaswani et al., 2017)



Linguistically-Informed Self-Attention for Semantic Role Labeling (Strubell et al., 2018)

Conclusion

NLP is a booming area: a lot of companies are interested in it, and many “big players” are involved (Google, Microsoft, Amazon, IBM, …)

It is a discipline which can connect to many other fields (such as computer vision)

If you are interested, we have plenty of projects and thesis

Paolo *dot* Torroni *at* unibo *dot* it

A *dot* Galassi *at* unibo *dot* it