Analyzing Text with Neural Networks

MA8701 Advanced statistical methods in inference and learning

Samia Touileb and Jeremy Barnes

Who are we?

Postdoctoral Fellow in the Language Technology Group, at the University of Oslo. I work on information extraction, sentiment analysis, and applications of machine learning to tasks within social science research.

I'm a Postdoctoral Fellow in the Language Technology Group at the University of Oslo. I work on neural network approaches to sentiment analysis and emotion detection.

Quick question for you

Quick question for you

What do you know about NLP?

What is NLP?

NLP?

- Computer speech and language processing
- Language engineering
- Human language technology
- Language technology
- Computational linguistics
- Natural Language Processing (NLP)

Get computers to perform tasks involving human languages

NLP?

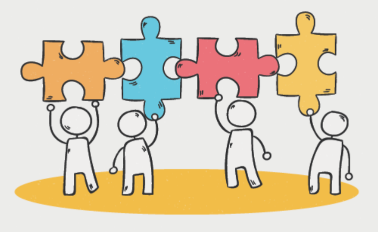
- Computer speech and language processing
- Language engineering
- Human language technology
- Language technology
- Computational linguistics
- Natural Language Processing (NLP)

Get computers to perform tasks involving human languages

Objective: *make computers "understand" natural languages*

NLP?

- Linguistics
- Informatics
- Statistics
- Machine learning
- Logic
- Philosophy
- Psychology
- ...



https://www.dpconline.org/images/DPC/Blog/WDPD2019/MAson_1.png

Very short history of NLP

- $\rightarrow$ – *2000*:
  - hand-crafted rule-based systems
  - probabilistic and data-driven models
- *2000 – 2010s*:
  - more data, more computing power $\rightarrow$ statistical machine learning
- *2010s – 2021 – ?*:
  - even more data, even more computing power $\rightarrow$ deep learning

Very short history of NLP

Computers can do anything!

Very short history of NLP

Computers can do anything!

- Machine translation was going to be solved in 4-5 years back in the 1950s

Very short history of NLP

Computers can do anything!

- Machine translation was going to be solved in 4-5 years back in the 1950s

But then ...

Very short history of NLP

Computers can do anything!

- Machine translation was going to be solved in 4-5 years back in the 1950s

But then ... Language

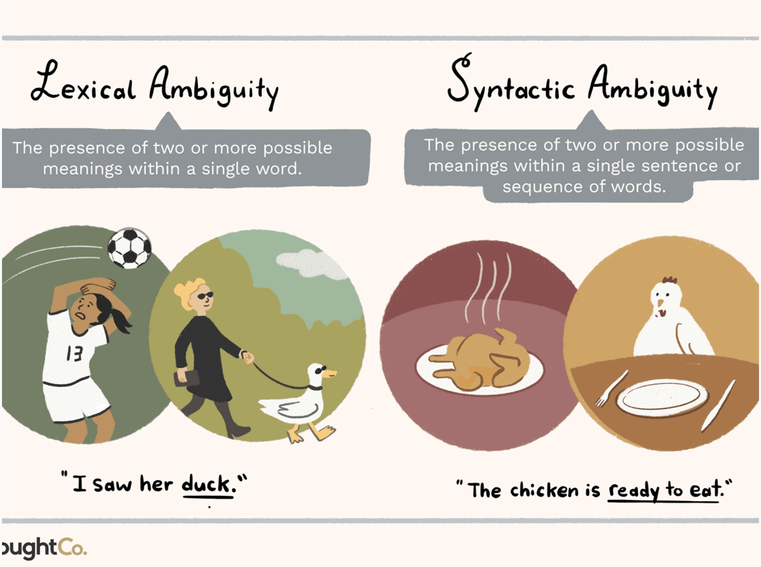
Why is language so challenging?

- Language is vague, different interpretations possible.
- Ambiguity everywhere.
- Provides compact communication: The same expression can be used in different contexts.

The ambiguities are largely invisible to us, we find the intended interpretation almost unconsciously.

For machines it is the opposite: easy to find all possible interpretations, but difficult to see which one is correct.

Ambiguity – lexical and structural



https://www.thoughtco.com/thmb/eYyVoPbyZkZ9JVNnSuqkimjeKOc=/1333x1000/smart/filters:no_upscale()/ambiguity-language-1692388_FINAL-dd68c7d1dd374507aa633d27539f0e62.png

Ambiguity

- I ate pizza with friends.
- I ate pizza with olives.

Ambiguity

- I ate pizza with friends.
- I ate pizza with olives.
- Friends and I shared some pizza.
- We shared some pizza.

Humans interpret linguistic expressions based on shared background knowledge and mutual expectations in a given context.

Understanding language == disambiguation.

- NLP: try to find strategies for how machines can cope with this.

Why is language so challenging?

Natural languages are *discrete*, *compositional*, and *sparse*

Why is language so challenging?

Language is *discrete*

- characters as basic elements.
- characters form words denoting objects, concepts, event, actions, ideas, ...
- characters and words are discrete symbols.

Why is language so challenging?

Language is *compositional*

- letters form words, words form phrases and sentences.
- meaning of a phrase is larger than the meaning of its individual words.
- meaning follows rules (*e. g.* negation).
- to interpret texts: need to analyse longer sequences than letters and words (phrases, sentences, documents).

Why is language so challenging?

Discreteness and compositionality leads to *sparsity*. Language is therefore *sparse*.

- words can be combined infinitely to to form meaning.
- infinite number of possible sentences.
- infinite (?) growth in vocabulary.
- not all words are known == no meaning:
  - no clear generalization from one sentence to another.
  - difficult to define similarity between sentences.

Why is language so challenging?

Discreteness and compositionality leads to *sparsity*. Language is therefore *sparse*.

- words can be combined infinitely to to form meaning.
- infinite number of possible sentences.
- infinite (?) growth in vocabulary.
- not all words are known == no meaning:
    - no clear generalization from one sentence to another.
    - difficult to define similarity between sentences.

No matter how huge the set of words, we are bound to observe new words, that are very different from all the known words.

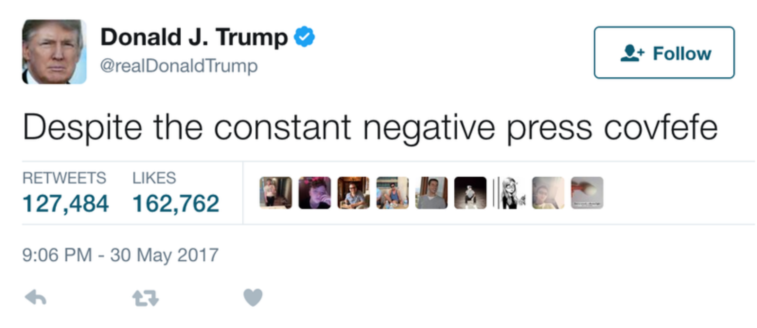
Why is language so challenging?

Language $\neq$ English

- “small” languages
- under-resourced languages (*e. g.* Norwegian)
- (spoken) dialects
- different scripts (Latin, Chinese, Arabic, Devanagari)

Why is language so challenging?

A lot of noisy human-generated data



https://media.npr.org/assets/img/2017/05/31/covfefe-trump_custom-2f50bc17c296cd744346c9b0626712bd7336caea-s800-c85.png

Preprocessing

Why preprocessing?

1. Break strings into logically sized pieces.
2. Convert these pieces into some representation that is useful for a computer.

Preprocessing

Tokenization

- *Word segmentation*.
- Separating out (tokenizing) words from running text.
- Easy for English: use whitespaces.
  - What about "New York"? "I'm"? "rock 'n' roll"?
- Still a major problem for other languages, *e. g.* Arabic.

Preprocessing

Sentence tokenization

- Generally based on punctuation.
- *e. g.* periods, question marks, exclamation points mark sentence boundaries.
- What about “Mr.”? “Inc.”?
- Rule-based or ML systems.

Preprocessing

- **Lemmatization**: sang, sung, sings $\rightarrow$ sing.
- **Lowercase**: Here, here.
- **Language identification**: Bokmål VS Nynorsk.
- **Normalization**: *e. g.* numbers. *1* fille et *1* garçon == *une* fille et *un* garçon.

Feature extraction

We have "cleaned" the texts. What is the next step?

Feature extraction

We have “cleaned” the texts. What is the next step?

Identify features!

Feature extraction

We have “cleaned” the texts. What is the next step?

Identify features!

Sentiment Analysis

“Extreme Ops” exceeds expectations.	POSITIVE
The actors are fantastic.	POSITIVE
Familiar but utterly delightful.	POSITIVE
Duvall is strong as always.	POSITIVE
This isn’t a new idea.	NEGATIVE
An absurdist spider web.	NEGATIVE
The movie is well done, but slow.	NEGATIVE
As it is, it’s too long and unfocused.	NEGATIVE

Feature extraction

Language Identification

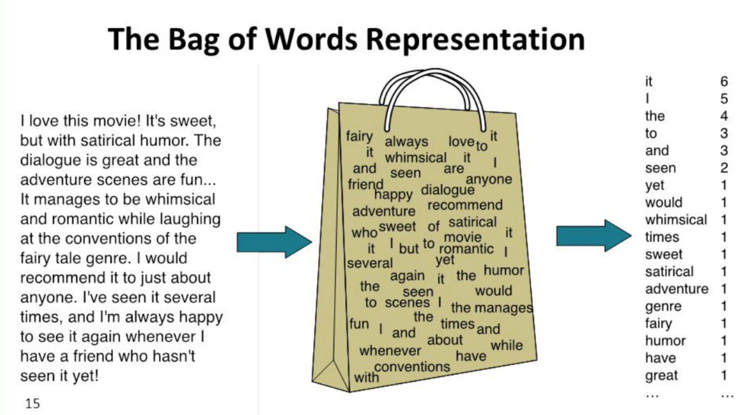
Idag er ho her. Etter to måneder. Velkommen sol.	Norwegian
Kan du huske, hvornår du kom på Twitter?	Danish
Twitter! Jeg har laga ramen!	Norwegian
Kender du en ven, der har det på samme måde?	Danish
eg skilji ikki hví Canal+ vísir Arsenal-Man	Faroese
Why did this make me emotional? That's so nice	English

Preprocessing

Data representation

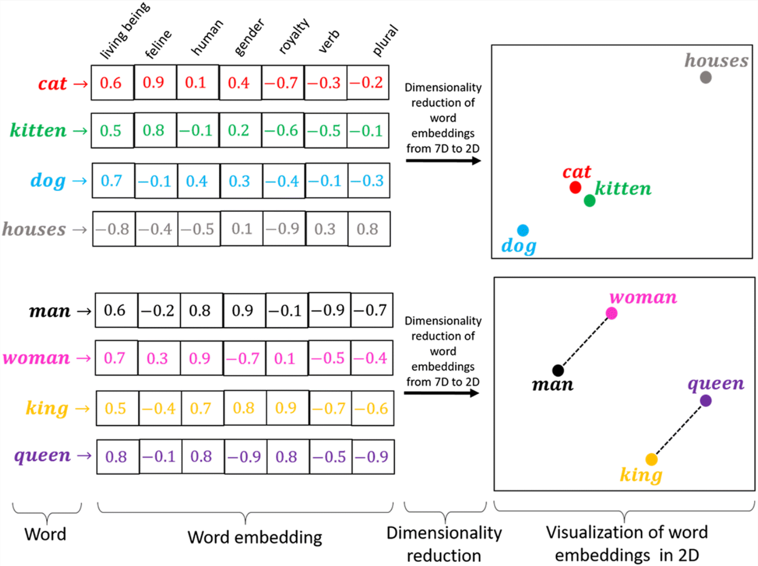
- BOW (bag-of-words).
- Word vectors.

Preprocessing – BOW



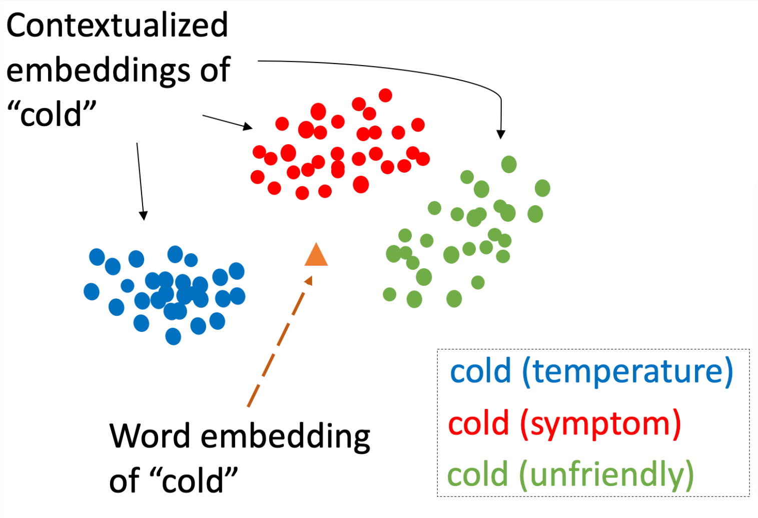
https://miro.medium.com/max/2552/1*MeSYCKGDOdwkJKVZKxJuvg.png

Preprocessing – word vectors



https://miro.medium.com/max/2598/1*sAJdxEsDjsPMioHyzlN3_A.png

Preprocessing – word vectors



https://images.deepai.org/converted-papers/1902.08691/pics/figure1.png

Recurrent Neural Networks

Why aren't BOW representations for text enough?

Why aren't BOW representations for text enough?

What would a bag-of-words representation look like for these examples?

Why aren't BOW representations for text enough?

What would a bag-of-words representation look like for these examples?

It was great , not a problem , only fun	Positive

Why aren't BOW representations for text enough?

What would a bag-of-words representation look like for these examples?

It was great , not a problem , only fun	Positive
It was not great , only a problem , not fun	Negative

Why aren't BOW representations for text enough?

What would a bag-of-words representation look like for these examples?

It was great , not a problem , only fun	Positive
It was not great , only a problem , not fun	Negative
It was not only great , it was fun , not a problem	Positive

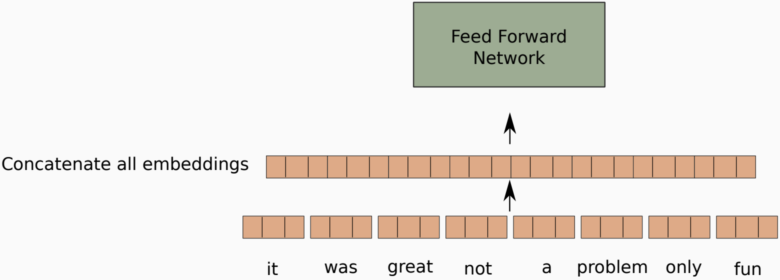
Why aren't BOW representations for text enough?

What would a bag-of-words representation look like for these examples?

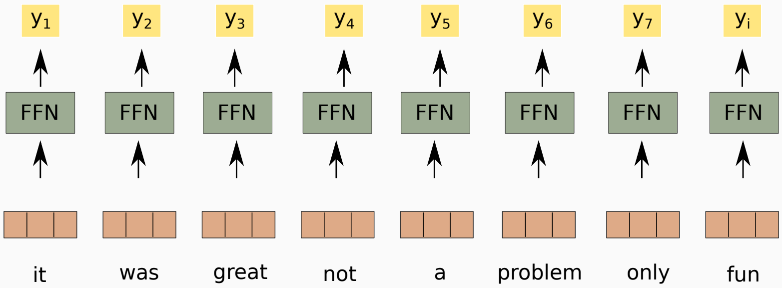
It was great , not a problem , only fun	Positive
It was not great , only a problem , not fun	Negative
It was not only great , it was fun , not a problem	Positive

···	the	a	it	is	was	fun	bad	great	problem	not	,	.	···
···	0	1	1	0	1	1	0	1	1	1	1	0	···
···	0	1	1	0	1	1	0	1	1	1	1	0	···
···	0	1	1	0	1	1	0	1	1	1	1	0	···

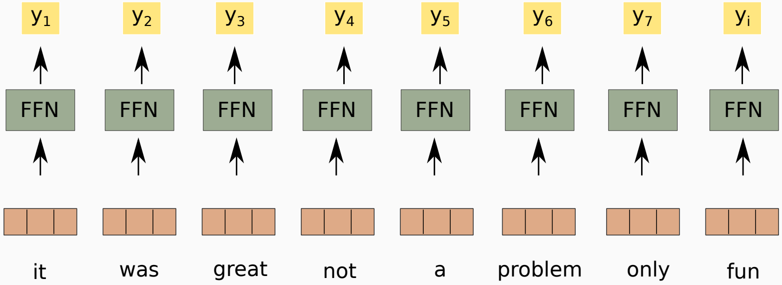
Similarly, Feed Forward Networks do not deal well with sequences of variable length.



Is it possible to unroll FFNs across time?

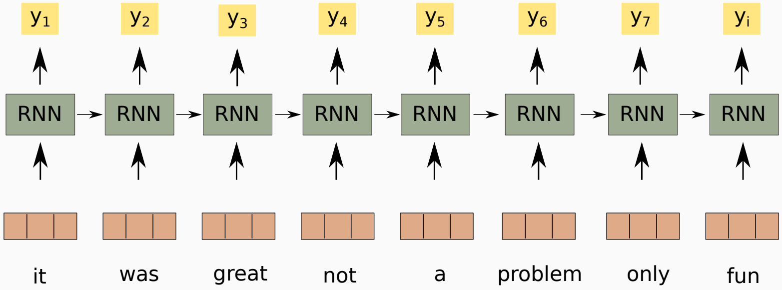


Is it possible to unroll FFNs across time?

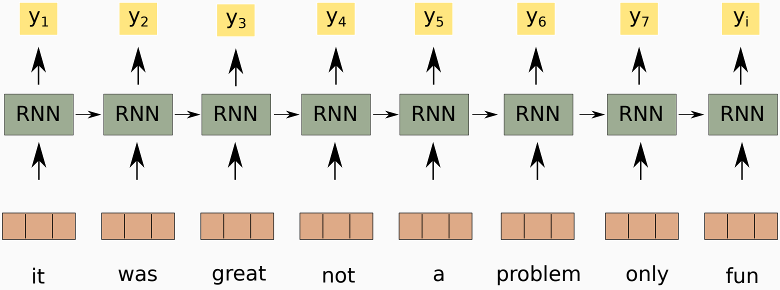


$$s_i = g(x_i W^x + b)$$
$$y_i = s_i$$

Is it possible to unroll FFNs across time?



Is it possible to unroll FFNs across time?



$$s_i = g(\underline{s_{i-1}W^s} + x_i W^x + b)$$
$$y_i = s_i$$

How do Recurrent Neural Networks fit text?

How do Recurrent Neural Networks fit text?

- Text and speech are sequential in nature.

How do Recurrent Neural Networks fit text?

- Text and speech are sequential in nature.
- Language is not.

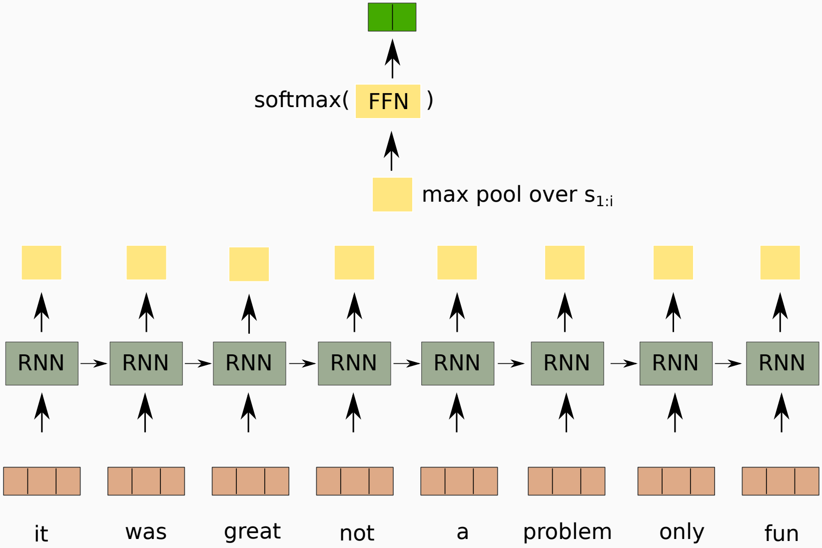
How do Recurrent Neural Networks fit text?

- Text and speech are sequential in nature.
- Language is not.
- Language can have both local- and long-distance dependencies and is more often thought of as hierarchical in nature.

How do Recurrent Neural Networks fit text?

- Text and speech are sequential in nature.
- Language is not.
- Language can have both local- and long-distance dependencies and is more often thought of as hierarchical in nature.
- However, an approximation that takes sequence information into account is better than one that ignores it

Example of RNN for sentiment classification



Practical problems with Elman RNNs

Practical problems with Elman RNNs

- Hard to train.

Practical problems with Elman RNNs

- Hard to train.
- exploding / vanishing gradient.

Practical problems with Elman RNNs

- Hard to train.
- exploding / vanishing gradient.
- in practice no one uses them for NLP tasks.

LSTMs and GRUs: better building blocks

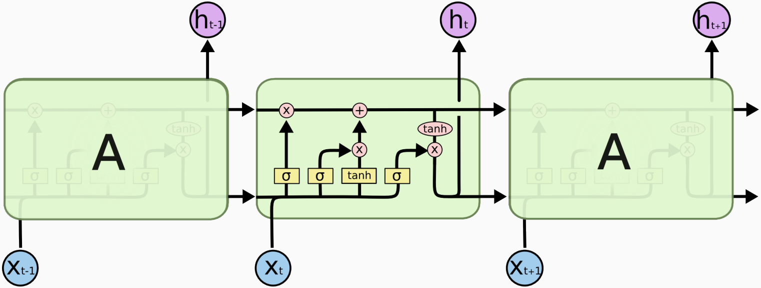
LSTM	GRU
$s_j = [c_j; h_j]$ $c_j = f \odot c_{j-1} + i \odot z$ $h_j = o \odot tanh(c_j)$	$s_j = (1 - z) \odot s_{j-1} + z \odot \tilde{s}_j$

LSTMs and GRUs: better building blocks

LSTM	GRU
$s_j = [c_j; h_j]$	$s_j = (1 - z) \odot s_{j-1} + z \odot \tilde{s}_j$
$c_j = f \odot c_{j-1} + i \odot z$	
$h_j = o \odot tanh(c_j)$	

c_j = memory cell	z = interpolation gate
h_j = hidden state	r = previous state access gate
f = forget gate	
i = input gate	
o = output gate	
z = update candidate	$\tilde{s}_j$ = update candidate

LSTM cell



https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-chain.png

Hands on example

Now, let's look at small example for sentiment analysis.

Hands on example

Now, let's look at small example for sentiment analysis.

Clone the github repository

```
git clone https://github.com/jerbarnes/MA8701-NTNU-2021-10
cd MA8701-NTNU-2021-10
```

Hands on example

First, let's look at the training data in the `data` folder.

Hands on example

First, let's look at the training data in the `data` folder.

We have three training sets:

Hands on example

First, let's look at the training data in the `data` folder.

We have three training sets:

Training sets
`train_small.tsv`
`train_medium.tsv`
`train_large.tsv`

Hands on example

We will be training two models: a Support Vector Machine trained with Bag-of-Words representations and a simple LSTM.

Hands on example

The `train_models.py` script has several arguments:

arguments

```
jeremy@Jeremy:~/Teaching/MA8701-NTNU-2021/code$ python3 train_models.py -h
usage: train_models.py [-h] [--training_epochs TRAINING_EPOCHS]
                       [--batch_size BATCH_SIZE]
                       [--embedding_dim EMBEDDING_DIM]
                       [--hidden_dim HIDDEN_DIM]
                       [--training_data TRAINING_DATA]

optional arguments:
  -h, --help            show this help message and exit
  --training_epochs TRAINING_EPOCHS
  --batch_size BATCH_SIZE
  --embedding_dim EMBEDDING_DIM
  --hidden_dim HIDDEN_DIM
  --training_data TRAINING_DATA
```

Hands on example

First, we'll train the models with only 500 examples. We'll run an interactive session so that we can play with the trained models.

python command

```
python3 -i train_models.py
```

Hands on example

First, we'll train the models with only 500 examples. We'll run an interactive session so that we can play with the trained models.

python command

```
python3 -i train_models.py
```

1. What is the difference in speed?
2. What kinds of errors do the two models make on the small test set?

Quick group session

We'll put you in breakout rooms for 5 minutes. Discuss the following questions with your colleagues.

Quick group session

We'll put you in breakout rooms for 5 minutes. Discuss the following questions with your colleagues.

Questions to answer in groups

1. What are some phenomena that the trained models are not able to deal with?
2. What kind of data could we use to improve this?
3. What kind of modeling assumptions could we include to improve this?

Conclusion

When working with text, you have to keep in mind how you are representing it.

When working with text, you have to keep in mind how you are representing it.

- How are you preprocessing the text? (sentences, tokens, subtokens?)

When working with text, you have to keep in mind how you are representing it.

- How are you preprocessing the text? (sentences, tokens, subtokens?)
  - **mBERT/NoTRAM:** 'Denne g ##jeng ##en h ##å ##per at de sammen skal bid ##ra til å gi k ##vinne ##fo ##t ##ball ##en i Kristiansand et lenge etter ##len ##gte ##t l ##ø ##ft .'
  - **NorBERT:** 'Denne gjengen håper at de sammen skal bidra til å gi kvinne ##fotball ##en i Kristiansand et lenge etterl ##engt ##et løft .'

Have you thought about the relationship between your task, your data, and the model you've chosen?

Have you thought about the relationship between your task, your data, and the model you've chosen?

- Is your model adequately expressive for the task?

Have you thought about the relationship between your task, your data, and the model you've chosen?

- Is your model adequately expressive for the task?
  - For course-grained tasks with lots of data and large documents, e.g., binary document classification, often a simpler model is enough.

Have you thought about the relationship between your task, your data, and the model you've chosen?

- Is your model adequately expressive for the task?
  - For course-grained tasks with lots of data and large documents, e.g., binary document classification, often a simpler model is enough.
  - For shorter texts, situations where you lack data, or more complex tasks, you need to choose the model more carefully.

Have you thought about the relationship between your task, your data, and the model you've chosen?

- Is your model adequately expressive for the task?
  - For course-grained tasks with lots of data and large documents, e.g., binary document classification, often a simpler model is enough.
  - For shorter texts, situations where you lack data, or more complex tasks, you need to choose the model more carefully.
- BOW is a good baseline. If you include TF-IDF, even better.

Have you thought about the relationship between your task, your data, and the model you've chosen?

- Is your model adequately expressive for the task?
  - For course-grained tasks with lots of data and large documents, e.g., binary document classification, often a simpler model is enough.
  - For shorter texts, situations where you lack data, or more complex tasks, you need to choose the model more carefully.
- BOW is a good baseline. If you include TF-IDF, even better.
- For tasks that require more fine-grained linguistic knowledge, long-distance relationships, and awareness of word order, RNNs are a good start.

Contact info:

{samiat, jeremycb}@ifi.uio.no