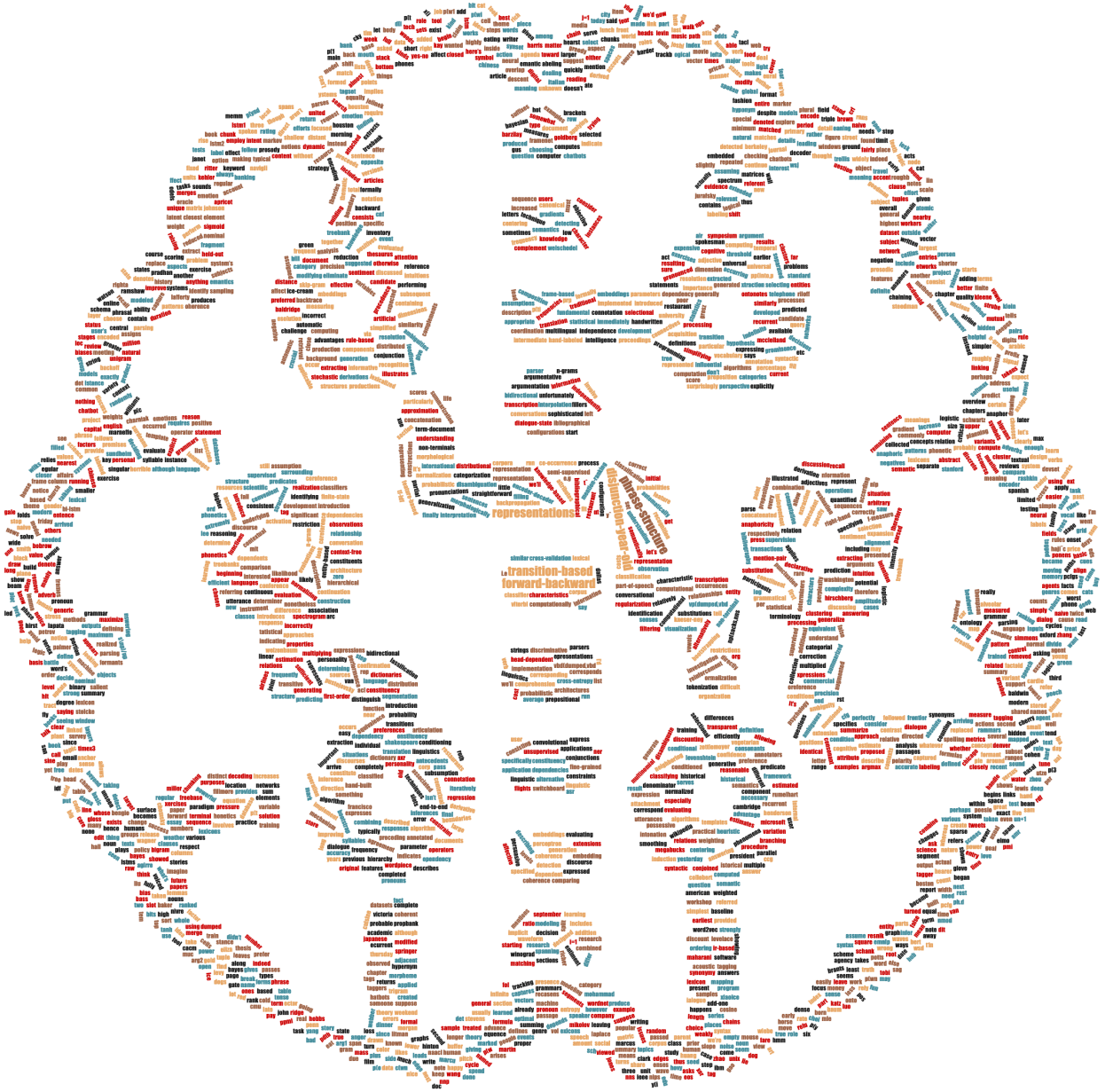
Representation Learning & Text Embeddings



Thomas Asikis,
asikist@ethz.ch
Department of Humanities, Social and Political Sciences, ETH Zurich

About Me

- BSc: Management - Economics - Computer Science
- MSc: Computer Science ~ Thesis: "*Part of speech tagging in Greek texts with word embeddings and deep neural networks*"
- PhD: Supporting Sustainable Development via Artificial Intelligence
- Professional Experience: Software Development, Analytics, Machine Learning…
- More on: https://tasikis.com/

Intro to Representations

Introduction

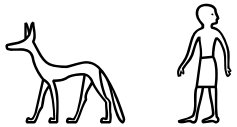
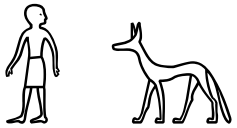
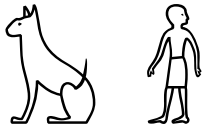
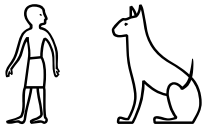
Contextual Structures

Representations

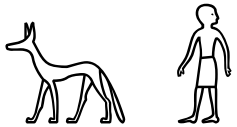
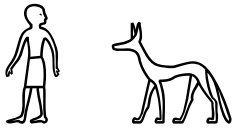
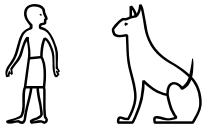
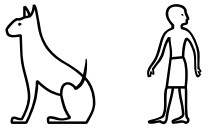
Tasks

Goals

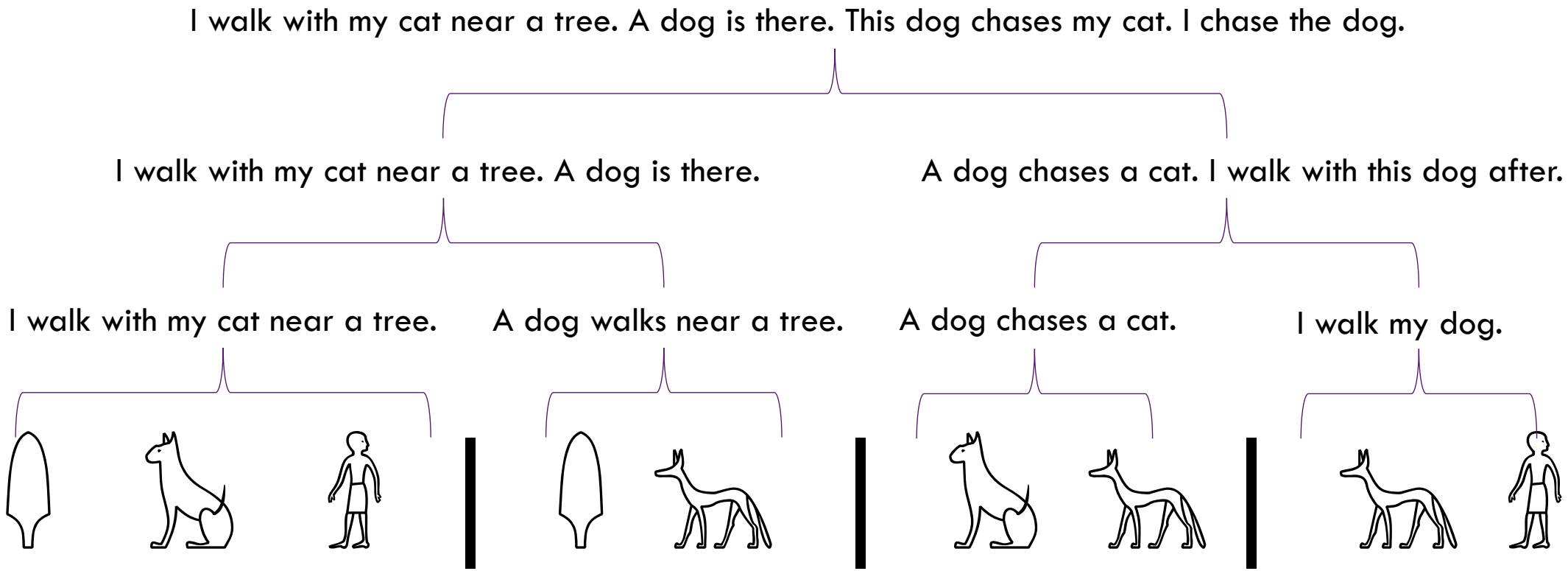
Sets

Symbols	Meaning
	I walk my dog
	I walk my dog
	?
	?

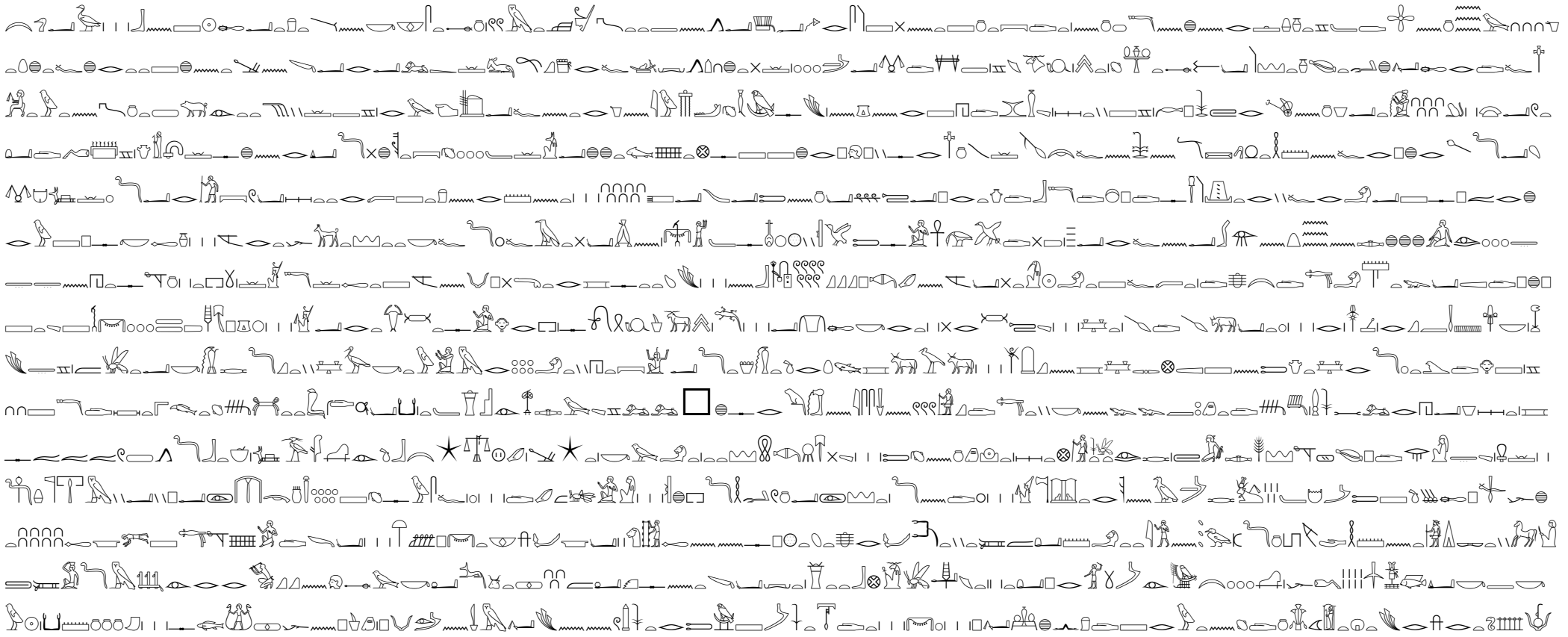
Sequences

Symbols	**Meaning**
	I walk my dog
	A dog chases me
	?
	?

Graphs


I walk with my cat near a tree. A dog is there. This dog chases my cat. I chase the dog.
I walk with my cat near a tree. A dog is there.
A dog chases a cat. I walk with this dog after.
I walk with my cat near a tree.
A dog walks near a tree.
A dog chases a cat.
I walk my dog.

Even more complex context structures...



Operation Representation

Symbols are nice and understandable, but can we construct an operationalized framework to capture and quantify underlying meanings?

-  $= [1,1]$
-  $= [10,10]$
-  $= [1,2]$
-  $= [1,1] + [1,2] = [2,3]$

Learning Representations

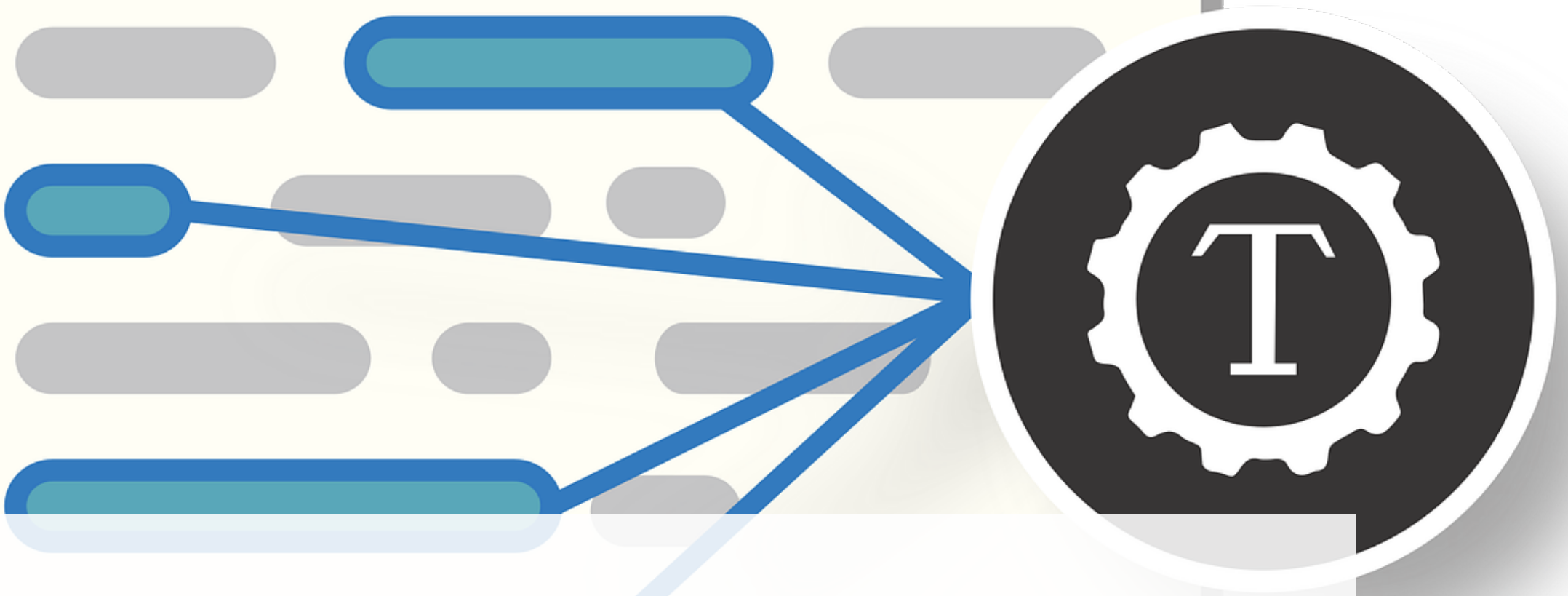
- Uniqueness: Different meanings require unique representations.
- Modelling: Semantic operations on meanings (abstraction, composition etc.) have to be possible on the representations as well.
- Informative: Representation needs to contain all information for a meaning.
- Computational considerations: Uniqueness & modelling can be achieved in many ways, but some are more efficient than others, e.g.:
  - 🦒 ← dog,
  - 🦒 ← $[0,1]$,
  - 🦒 ← 10^{100}

Structures of Words

Although most of the times we focus on words, the research goal might require to process other language structures:

- Characters compose words.
- Words compose sentences.
- Sentences compose paragraphs.
- Paragraphs compose documents.
- Documents compose corpora.

As representations can be constructed for words, they can also be constructed for all other structures as well.

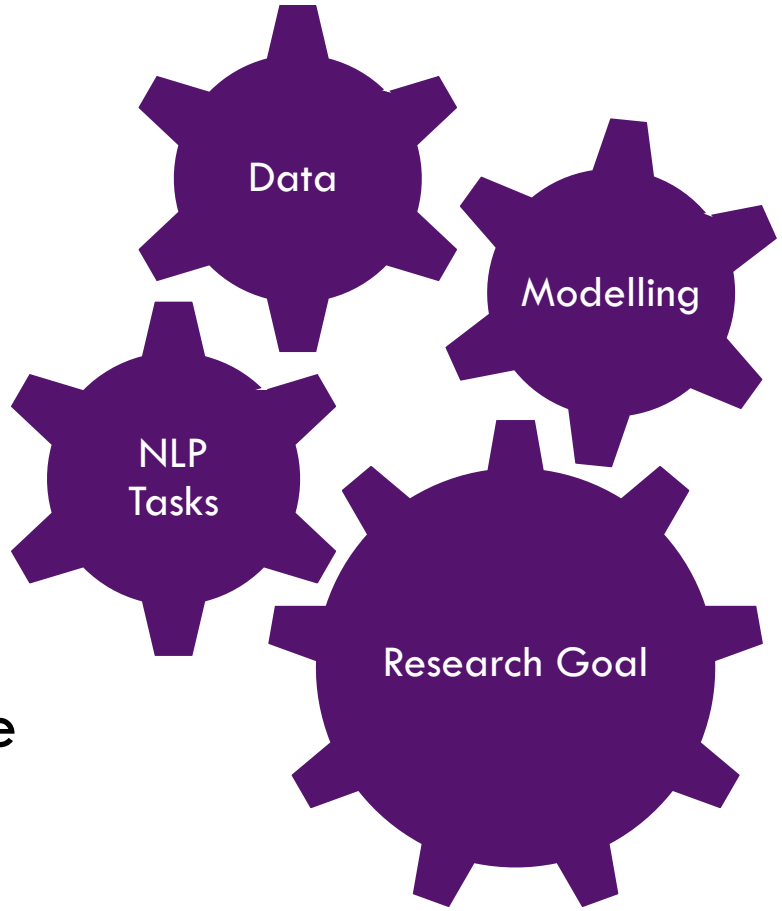


Text Processing for Research

A Quick Setup

Everything is interconnected:

- Define a research goal: e.g. GAL-TAN position of party based on their statements.
- Determine available data. E.g. parliamentary corpora.
- Pick a set of NLP tasks that contribute to explaining this goal. E.g. Text Regression on GAL-TAN scale.
- Choose a model that is compatible with data, captures contextual operations required for the NLP tasks and the goal.



Decide on the Research Goal

- Point out the difference of parties based on their parliamentary texts.
- How would I do it without the parliamentary texts?
- How would I do it if I studied all the texts manually?
- Can I find analogues and common concepts between the non-text or manual methods?
- Can I break down the analogues/common concepts to NLP tasks?

Choosing Relevant NLP Tasks

Decide on:

- the structures that I need to evaluate, e.g. words, sentences, documents, collections of documents?
- the evaluation tasks, e.g. classification, regression, distance tasks (nearest neighbors, similarities), composition, pure representation?
- analysis on task results, e.g. distance comparison, class analysis, clustering etc.

Choosing the Model

- What assumptions are made for the task based on data? E.g. part of speech tagging on sentences is assumed to have an underlying tree contextual structure.
- What assumptions does my model satisfy? E.g. Hidden Markov Models address sequential dependencies.
- What are the computational considerations of the model?
  E.g. a non-linear SVM classifier has higher training times than a logistic regression!
- Models require specific preprocessing: e.g. W2V requires stop-word removal vs GloVe that doesn't. Can this preprocessing be done?
- Does the data quantity and quality satisfy the model requirements? E.g. avoid training neural models on small corpora.

Performance Evaluation

Which performance?

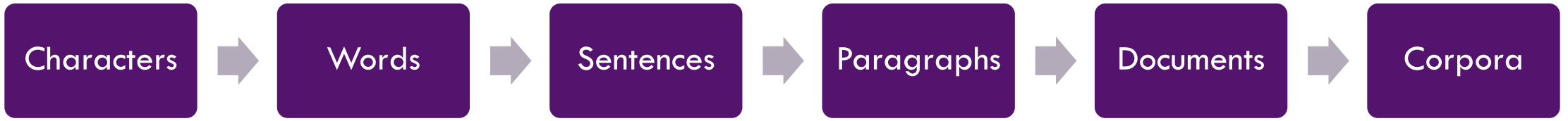
- Learning performance: How good does the network learn? Overfitting/Underfitting etc. Usually we look at the loss function.
- Task performance: How well am I doing in the task? There are specific evaluation metrics for each task, e.g. for classification accuracy.
- Goal performance: Does learning and achieving the task help my goal? E.g. are political parties significantly different based on my model outcome?

A Look on Data

Text Data Structures

Text data may have a lot of structures, but a general pattern can be compositional:

e.g.:



NLP Tasks and Data

- Most NLP tasks that can be defined for an element can be also defined for its compositions:
- E.g. distance of between **words** → distance between **corpora**.
- E.g. sentiment labels of **words** → sentiment label between **corpora**.

! Finding the right operation to aggregate the task results of an element (e.g. classify word) to the task of its collection (e.g. classify sentence) is not trivial.

The distance between corpora is not always the sum/mean between word distances. E.g. assume contextual distance between sentences even if you know the distance between words:

I drove to my parents.

I drove to my friends.

I walked to my father's house to meet him and my mom.

Tokenization

Split a complex collection of text elements to the individual element.

Goal: Iterate through individual elements to address a task.

Examples:

- Convert a corpus to a sequence of individual words.
- Convert a sentence to a sequence of individual characters.

! Tokenization may rely not only on delimiters but also on patterns.

E.g. saying tokenize on space " " and dot "." might tokenize an abbreviation (E.T.H.) to characters.

Sometimes tokens may be "invisible" and need to be inserted:
E.g. when a new paragraph starts, we prepend the token "#$#".

Normalization

Convert all text elements to a specific case and character form (e.g. remove accents). The goal is to avoid treating same elements as different due to capital/lowercase letters and accents.

E.g.

”Can I offer help?”, “I can assist you if you want.”

⇩ ⇩

”can i offer help?”, “i can assist you if you want.”

! Normalization may remove useful patterns. Again abbreviations or names may be affected.
E.g. CAN is a company that produces the best cans…

Filtering

Keep text elements that are of interest. Remove text elements that are of noise.

A very popular method is stop-word removal.

Another widely used method is keeping words from specific dictionaries.

E.g.

“The two US diplomats reached an agreement with the Iranian authorities.”

⇩

”US diplomats reached agreement Iranian authorities.”

Filtering techniques are extremely powerful when combined with high domain expertise and the tasks are simple.
Using them reduces also the complexity of models to use.

Dictionaries

- Collections of text elements of interest.
- May be a map from a word to a numerical structure: e.g. words mapped to sentiment classes/values/vectors.
- It is relevant to corpus characteristics and the research goal.
- Generating dictionaries can be done by experts but also assisted by NLP tasks, such as topic modelling.

Stemming

Remove affixes (suffixes, prefixes, infixes, circumfixes) of words to obtain a word stem.

This way words of different form but same meaning are mapped to the same text element.

Useful when a task focuses on general meaning behind a word.

e.g.:

“running”, “runner” → “run”

This might not always work: “worse” and ”bad”.

Lemmatization may work better in this case, but it is an NLP task on its own as it requires knowing the part of speech.

Collocation Detection

The habitual juxtaposition of a particular word with another word or words with a frequency greater than chance.

In many tasks, collocations can be detected from frequency of word co-occurrence (e.g. pairs of words that co-occur more than others).

Collocation can be joined and treated as a single text element during analysis.

Windows

Break corpus into windows of 5 words:

The quick brown fox jumps over the lazy dog:

Window 1: [The quick brown fox jumps]

Window 2: [quick brown fox jumps over]

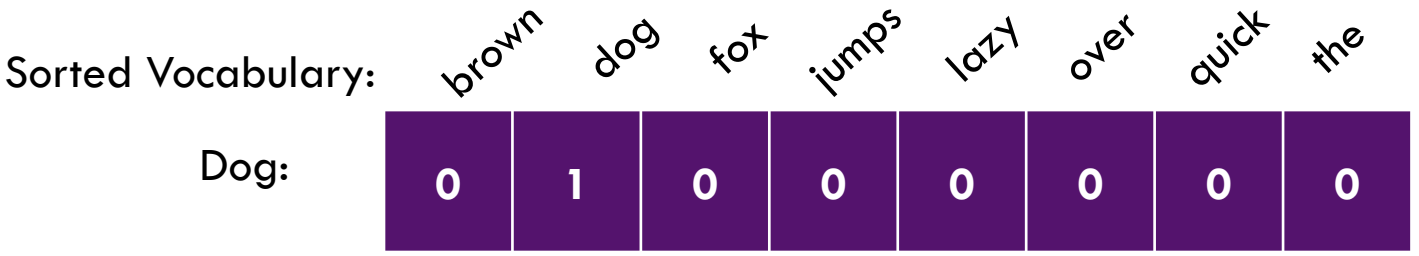
Window 3: [brown fox jumps over the]

Window 4: [fox jumps over the lazy]

Binary Representation

- Collect all unique words from a corpus: create a vocabulary.
- Sort them.
- Map each word to an index (an integer number indicating the position of the word in the sorting).
- For each word create a big vector filled with 0, with size equal to the words of the vocabulary.
- For each word and vector put 1 in the element of the vector corresponding to the word index.

Corpus: The quick brown fox jumps over the lazy dog



More Representations

Instead of binary vectors from vocabulary, we can map any numerical vector generated from a process from that word.

E.g.: term frequency over documents:

Documents	Doc. 1	Doc. 2	Doc. 3	Doc. 4	Doc. 5	Doc. 6
Dog:	0.3	0.5	0.2	0.1	0.3	0.2

Embeddings:

Dimensions	1	2	3	4	5	6
Dog:	0.3	-0.5	2.2	1.1	7.3	-1.2

Co-Occurrence matrix

Corpus: The quick brown fox jumps over the lazy dog

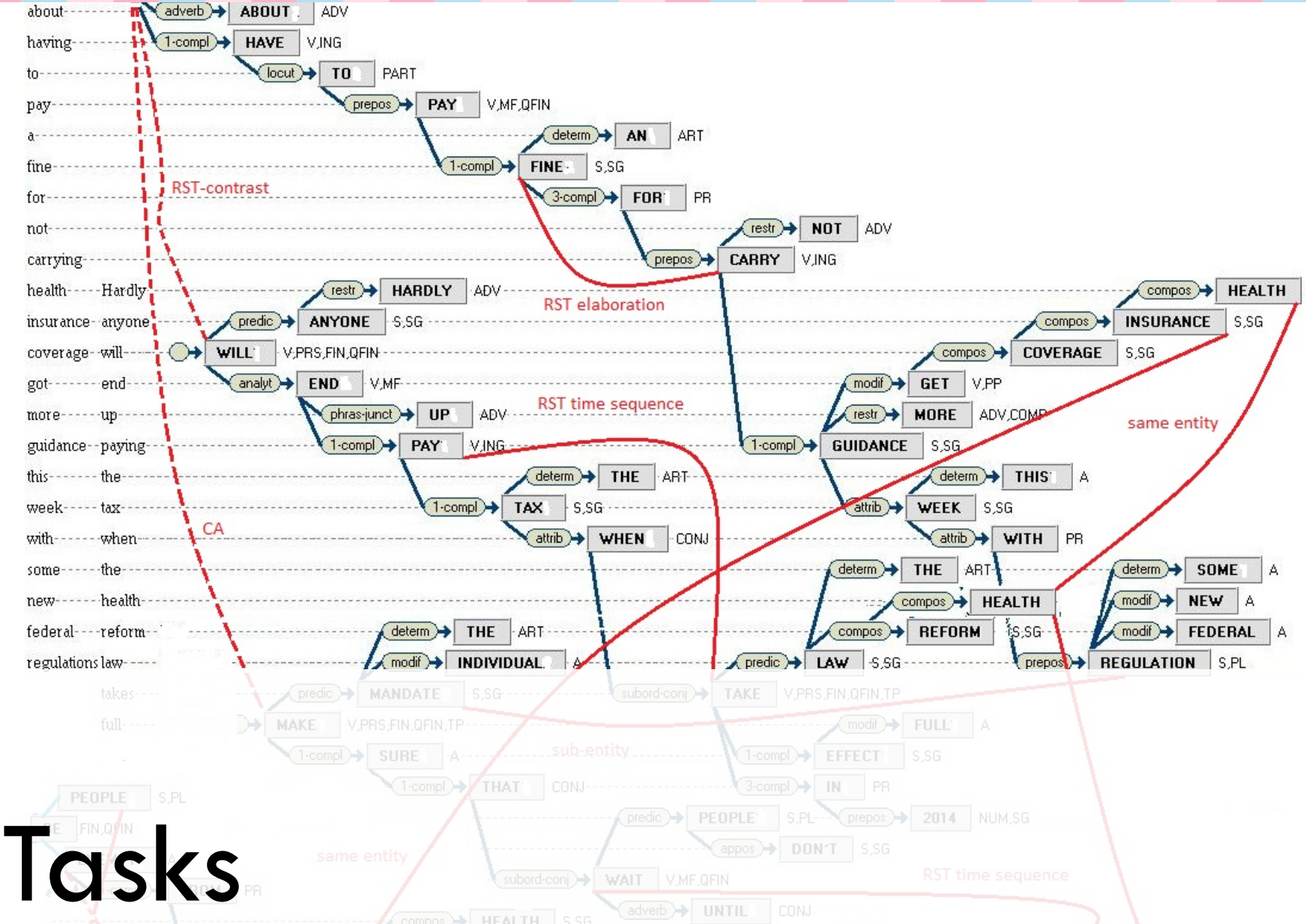
Window: 3 words

Words	the	quick	brown	fox	jumps	over	lazy	dog
the	0	1	1	1	0	0	1	1
quick	0	0	1	1	1	0	0	0
brown	0	0	0	1	1	1	0	0
fox	1	0	0	0	1	1	0	0
jumps	1	0	0	0	0	1	1	0
over	1	0	0	0	0	0	1	1
lazy	0	0	0	0	0	0	0	1
dog	0	0	0	0	0	0	0	0

Creating Data Supervised Tasks

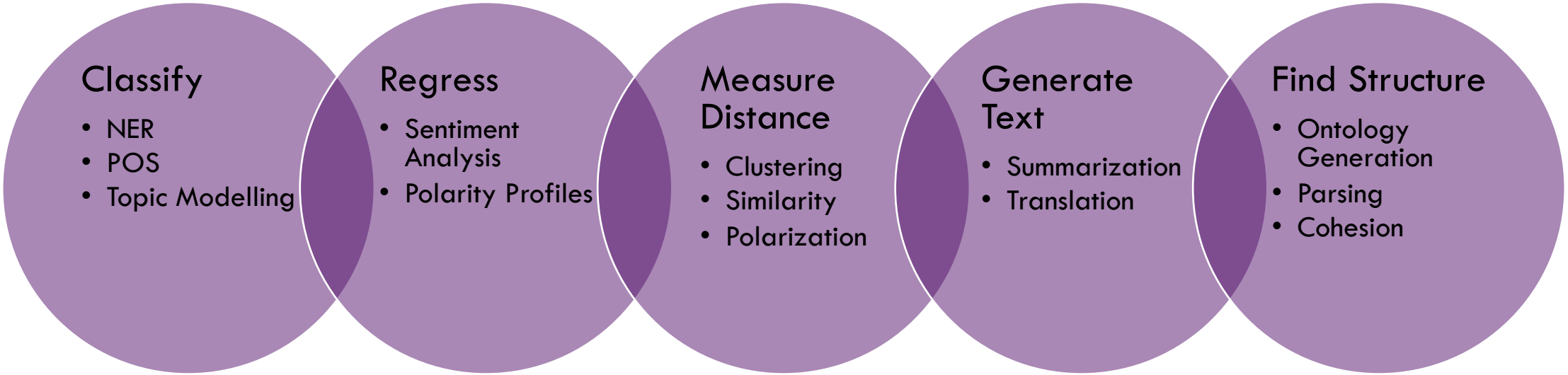
When we plan to train models that rely on expert data, several considerations arise:

- Inter-coding is important. Most models overfit, therefore it is very likely to learn coder bias.
- The designer of the experiment may have a higher bias when assigning labels.
- Check for imbalanced classes. Unless necessary, avoid creating imbalanced classification schemes.
- Check if your labels are crisp or fuzzy! Are labels always mutually exclusive?
- For NLP tasks creating a golden standard corpus (high quality prototypical corpus) for a novel task is a big contribution.


about
having
to
pay
a
fine
for
not
carrying
health
insurance
coverage
got
more
guidance
this
week
with
some
new
federal
regulations
Hardly
anyone
will
end
up
paying
the
tax
when
the
health
reform
law
adverb
ABOUT
ADV
1-compl
HAVE
V,ING
locut
TO
PART
prepos
PAY
V,MF,QFIN
determ
AN
ART
1-compl
FINE
S,SG
3-compl
FOR
PR
prepos
CARRY
V,ING
restr
NOT
ADV
RST-contrast
RST elaboration
restr
HARDLY
ADV
predic
ANYONE
S,SG
WILL
V,PRS,FIN,QFIN
analyt
END
V,MF
phras-junct
UP
ADV
1-compl
PAY
V,ING
RST time sequence
determ
THE
ART
1-compl
TAX
S,SG
attrib
WHEN
CONJ
CA
determ
THE
ART
modif
INDIVIDUAL
A
compos
HEALTH
compos
INSURANCE
S,SG
compos
COVERAGE
S,SG
modif
GET
V,PP
restr
MORE
ADV,COMP
1-compl
GUIDANCE
S,SG
same entity
determ
THIS
A
attrib
WEEK
S,SG
attrib
WITH
PR
determ
THE
ART
compos
HEALTH
compos
REFORM
S,SG
predic
LAW
S,SG
determ
SOME
A
modif
NEW
A
modif
FEDERAL
A
prepos
REGULATION
S,PL
takes
predic
MANDATE
S,SG
subord-conj
TAKE
V,PRS,FIN,QFIN,TP
full
MAKE
V,PRS,FIN,QFIN,TP
modif
FULL
A
1-compl
SURE
A
sub-entity
1-compl
EFFECT
S,SG
1-compl
THAT
CONJ
3-compl
IN
PR
PEOPLE
S,PL
predic
PEOPLE
S,PL
prepos
2014
NUM,SG
appos
DON'T
S,SG
same entity
subord-conj
WAIT
V,MF,QFIN
RST time sequence
adverb
UNTIL
CONJ
compos
HEALTH
S,SG

NLP Tasks

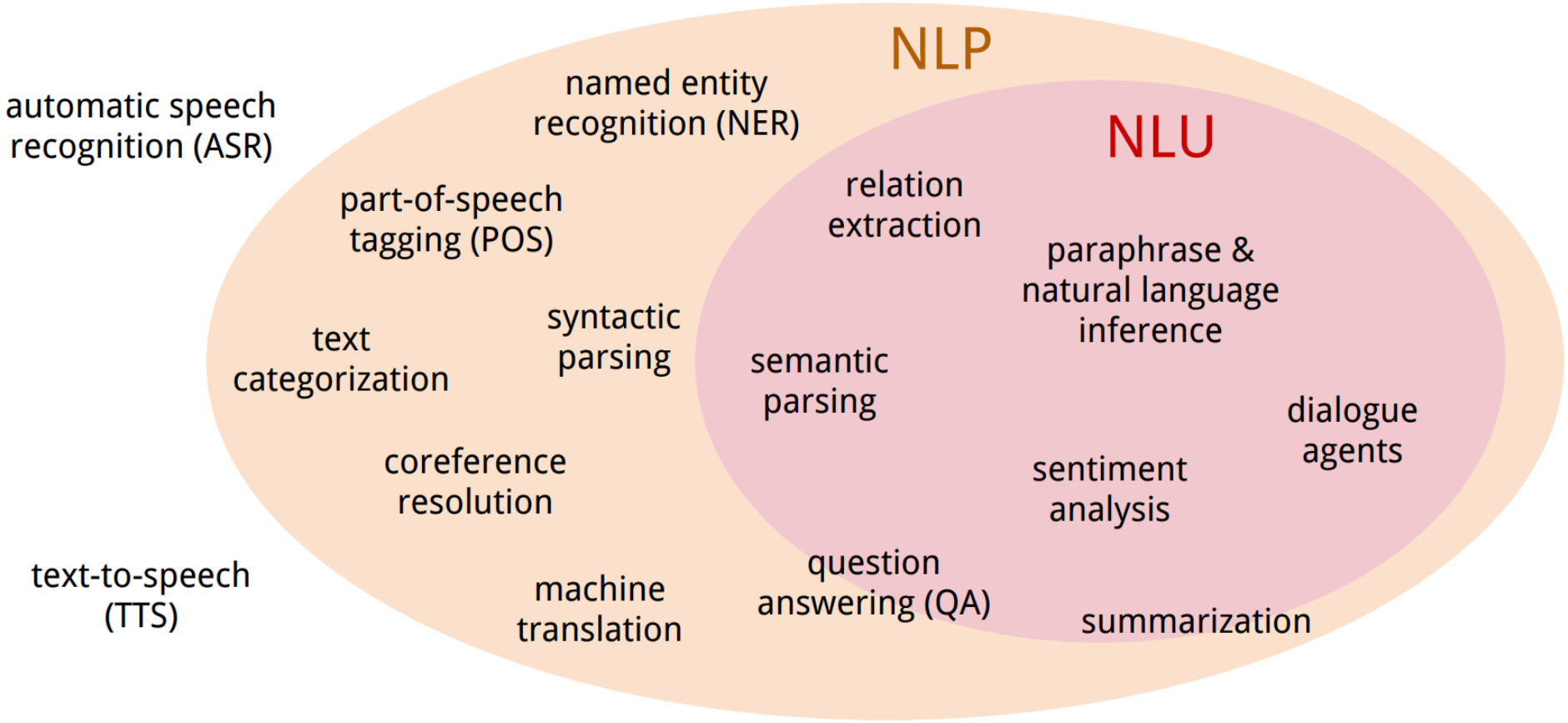
NLP Tasks: Method Classification



NLP Tasks: Characteristics

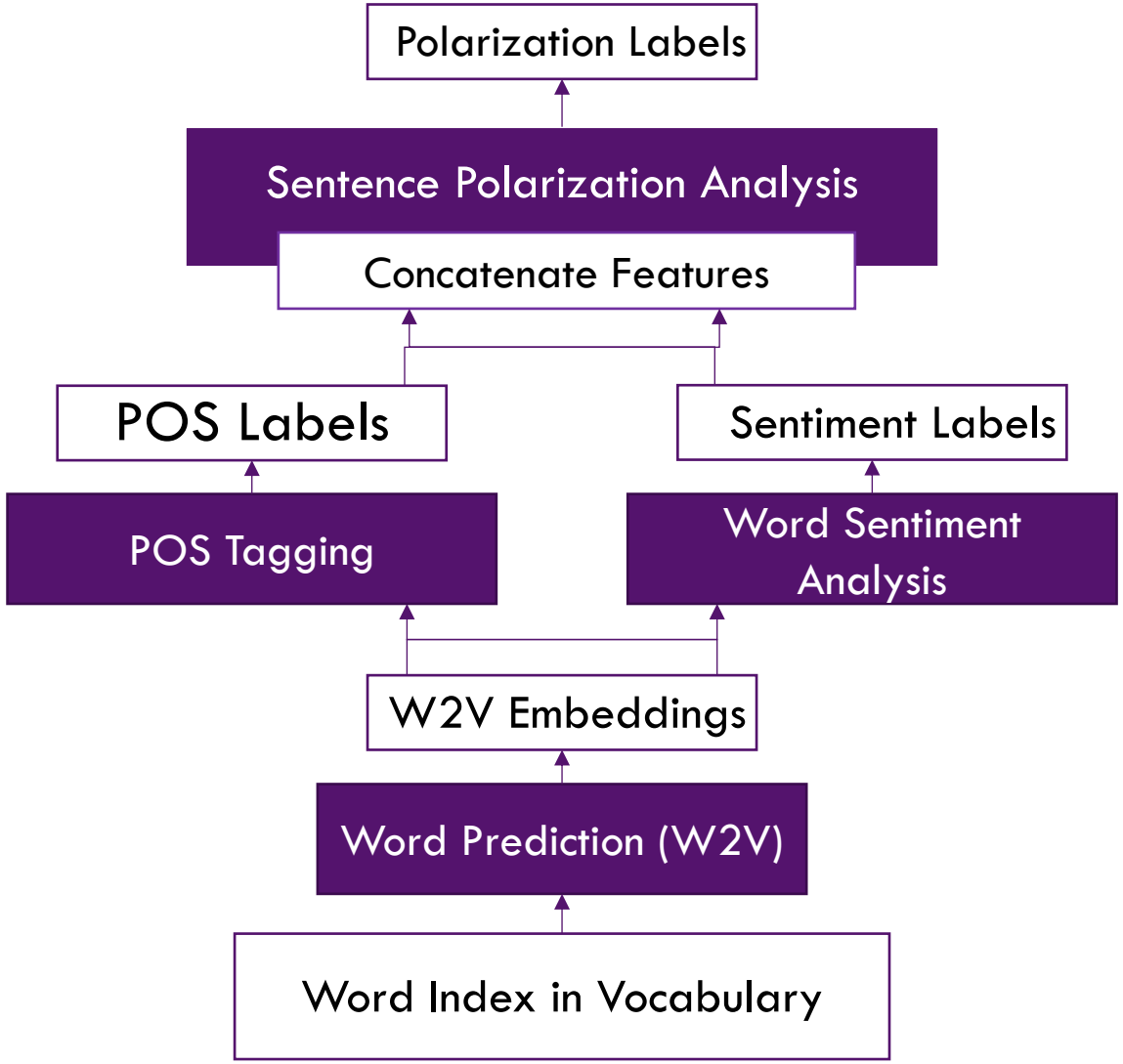
- Majority of tasks take text as input.
- Input may be combined with numbers and other information.
- Outputs vary: Text, numbers, labels etc.
- Representations can be input of tasks or learned through tasks.
  E.g. embedding of word learned to predict a label assigned to it.
- Task specific representations: Often capture underlying context of text elements in relation to the task.

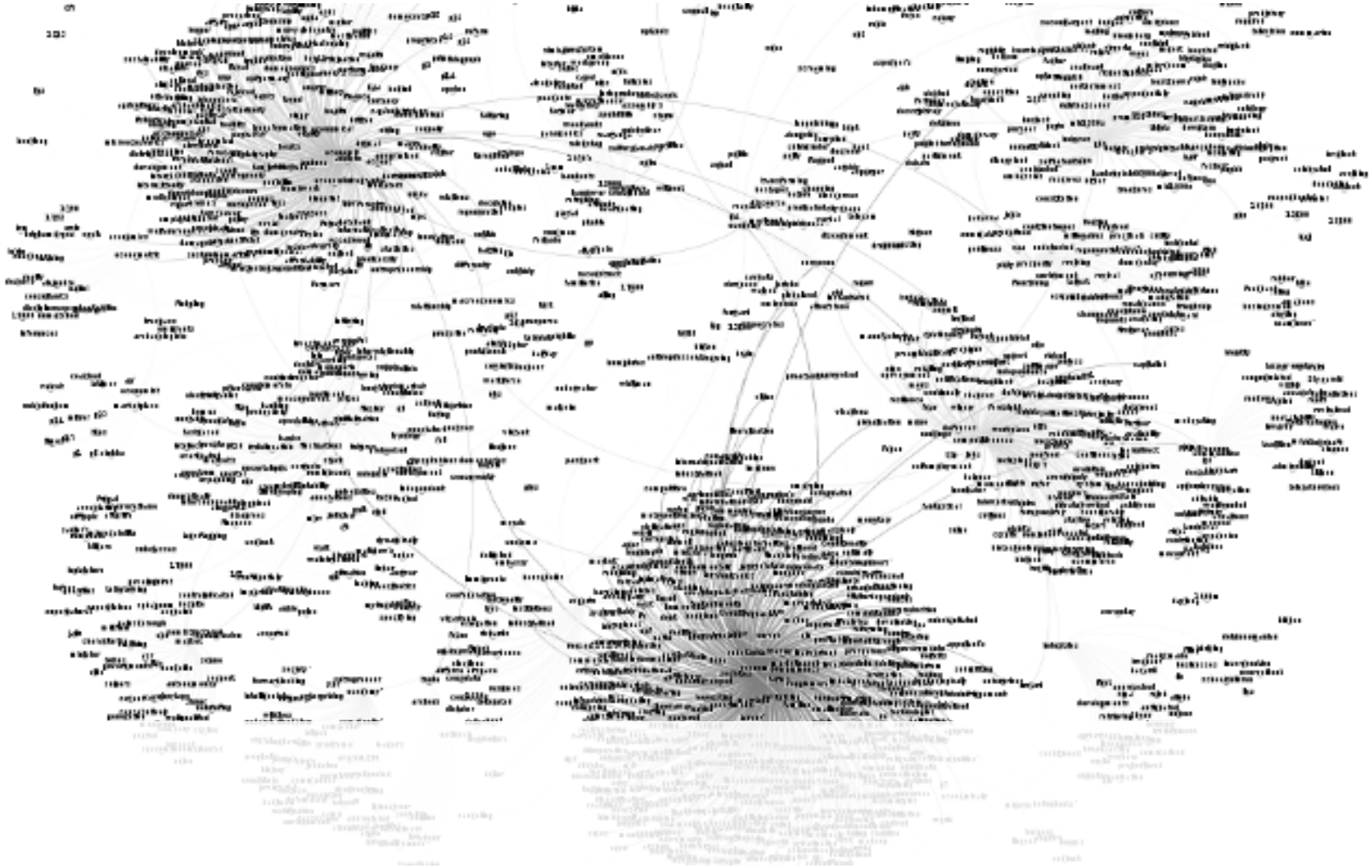
NLP-NLU Classification



https://nlp.stanford.edu/~wcmac/papers/20140716-UNLU.pdf

Task Composition





Representation Techniques

Representation Engineering: Count Vectors

- The main idea: Count occurrences of text elements (e.g. word frequencies).
- Bag of words: Set of words in word structure (e.g. document).
- Term frequencies:

$$\frac{\#\ of\ specific\ word\ appearences\ in\ document}{\#\ all\ words\ in\ document}$$

- Inverse document frequency:

$$\log\left(\frac{\#\ documents}{\#\ of\ documents\ with\ specific\ word}\right)$$

Latent Dirichlet Allocation

- Topic Modelling: Find topics or classes for clusters of documents.
- Latent Semantic Indexing/Analysis (LSI/LSA) → probabilistic LSI → LDA.
- Input: Document/Term Matrix.
- Main ideas:
  - Every document is a mixture of topics.
  - Every topic is a mixture of words.
  - The above mixtures can be learned via a probabilistic model.
  - Stochastic model: (Relies on sampling) results may differ per run… may suffer with small datasets.
  - Usually more accurate but also slower than LSI, pLSI.
- Interesting parameters: k number of topics.
- General method. Good to use always when starting analysis on new text data.
- **Topics and words can be used as input features for models → nominal (categorical) representations!**
- Word collections as sets.

Topic Representation

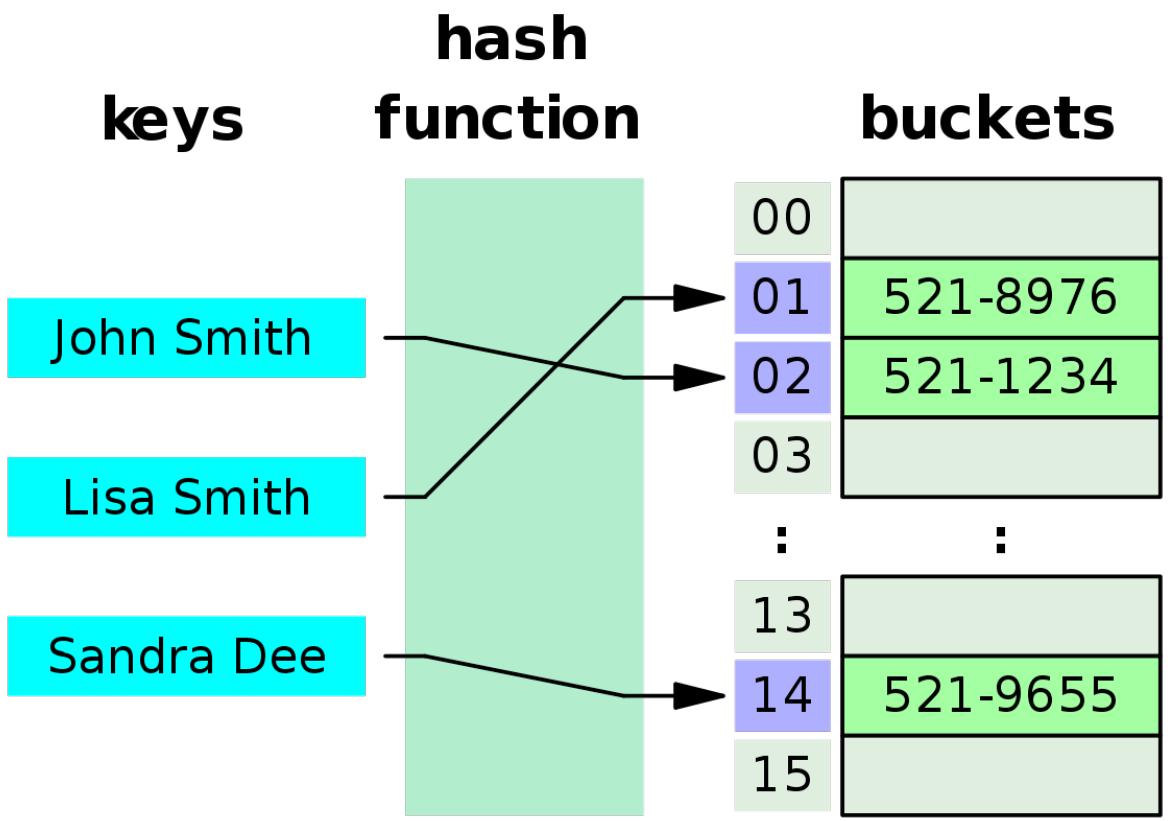
Topics	1	2	3	4	5	6
Document:	**0.3**	**-0.5**	**2.2**	**1.1**	**7.3**	**-1.2**

Hashing

Hashing:

- Produced by a hash function over an input (e.g. words).
- Buckets: The outcome of a hash function (an id and pointers to real data).
- Collisions: 2 different inputs get same bucket. If the inputs are similar this behavior is preferred.
- Implications: Using a hash function can help to operate fast on data.
- Idea: Collisions can put similar documents in same bucket.

Hash Functions and Buckets



https://en.wikipedia.org/wiki/Hash_table

Local Sensitive Hashing

- A possible problem: Get 20 most similar documents out of 2 billion…
- Inputs: A corpus with documents (just text).
- Outputs: Buckets.
- Parameters: Bands b.
- Applications: Approximate fast k-nearest neighbors search. Specific hash function → specific distance for k-NN (e.g. minhashing → Jaccard).
- Approximate: We can get 100 and then use normal distance on them to get 20.
- Higher b implies lower **similarity threshold (higher false positives) and lower b implies higher similarity threshold (higher false negatives)*.**
- Works best on extremely large datasets.
- Word collections as sets.
- Representation for each document**: Set of nearest neighbors or distance from fixed documents (characteristic documents).**

*https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134

Considerations

- The previous methods tell us how to find topics and match documents.
- Still, those representations are not relevant to local context.
- LDA uses tf-idf which assumes documents are sets of words.
- LSH uses shingling and then hashing, again assuming documents are sets of words.
- So any contextual or pragmatic operations on features from these methods assume contextual/logical relationships to arise from sets…
- In both methods the task is to find global non-sequential similarity.

Co-Occurrence Embeddings

Try to create a numerical representation that captures co-occurrence of words. This is assumed analogue to capturing local semantic context.

Notable methods:

- Word2Vec (via a neural network)
- GloVe (by factorization of co-occurrence matrix)

Word2Vec

- Goal: Predict a word from its context (or the other way around).
- Data: Binary representations for each word.
- Context: Windows of same size around words.
- Negative sampling - additional goal: Predict which words will never be in context (generalization, dataset augmentation).
- Localized context.

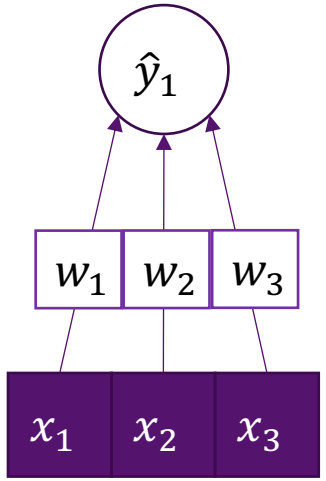
Introduction to Neural Methods

Let's illustrate a linear regression:

Goal: Find the values for w_1, w_2, w_3 so that the error is minimized over all samples. This optimization problem can be solved efficiently with many techniques. E.g. Newton's method.

Estimated Value:

$\hat{y}_1 = w_1 x_1 + w_2 x_2 + w_3 x_3$



y_1:Actual value

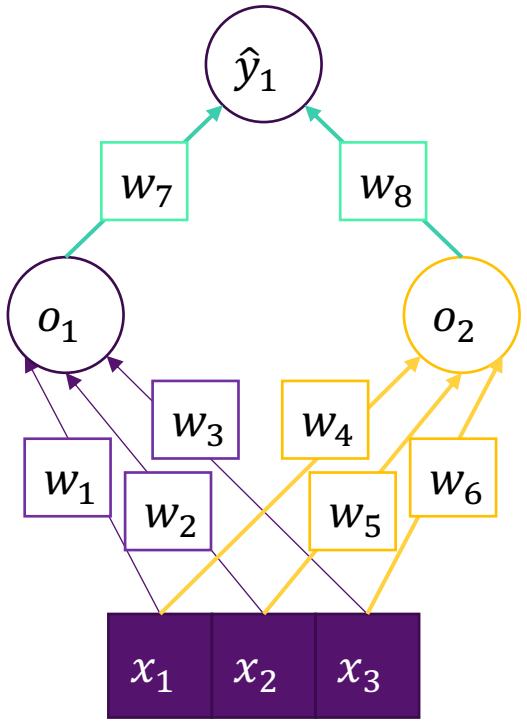
Error Calculation: $(y_1 - \hat{y}_1)^2$

Training sample:

Introduction to Neural Methods

So let's assume that the relationship between our data is a bit more complicated.

We change our model a bit:



Goal: Find the values for:

$w_1, w_2, w_3, w_4, w_5, w_6, w_7, w_8$

so that the error is minimized over all samples.

This optimization problem can be solved efficiently with some techniques.

Most famous: Stochastic gradient descent with backpropagation.

y_1:Actual value

Error Calculation: $(y_1 - \hat{y}_1)^2$

Estimated values:

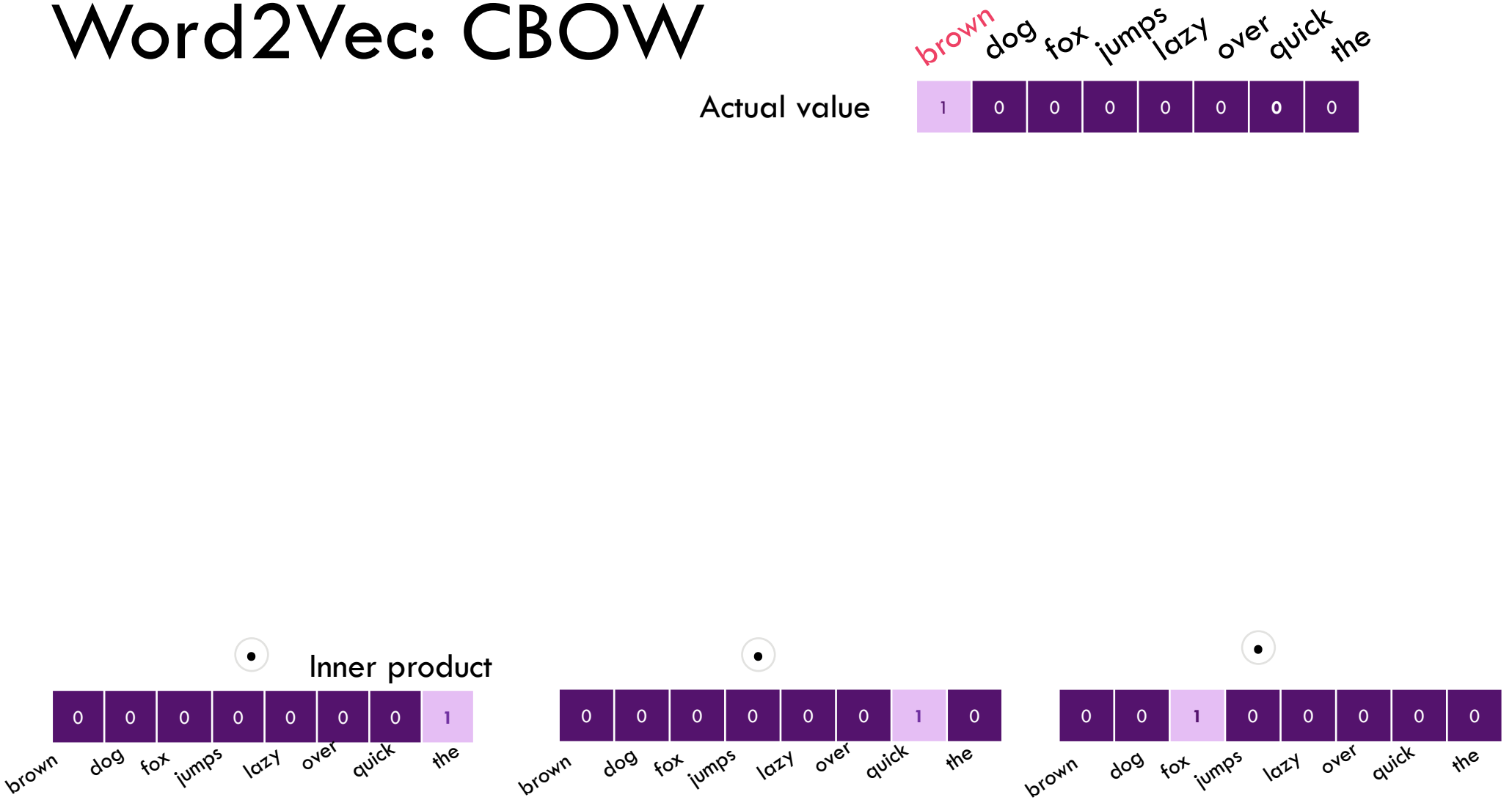
$o_1 = (w_1x_1 + w_2x_2 + w_3x_3)^2$

$o_2 = (w_1x_1 + w_2x_2 + w_3x_3)^3$

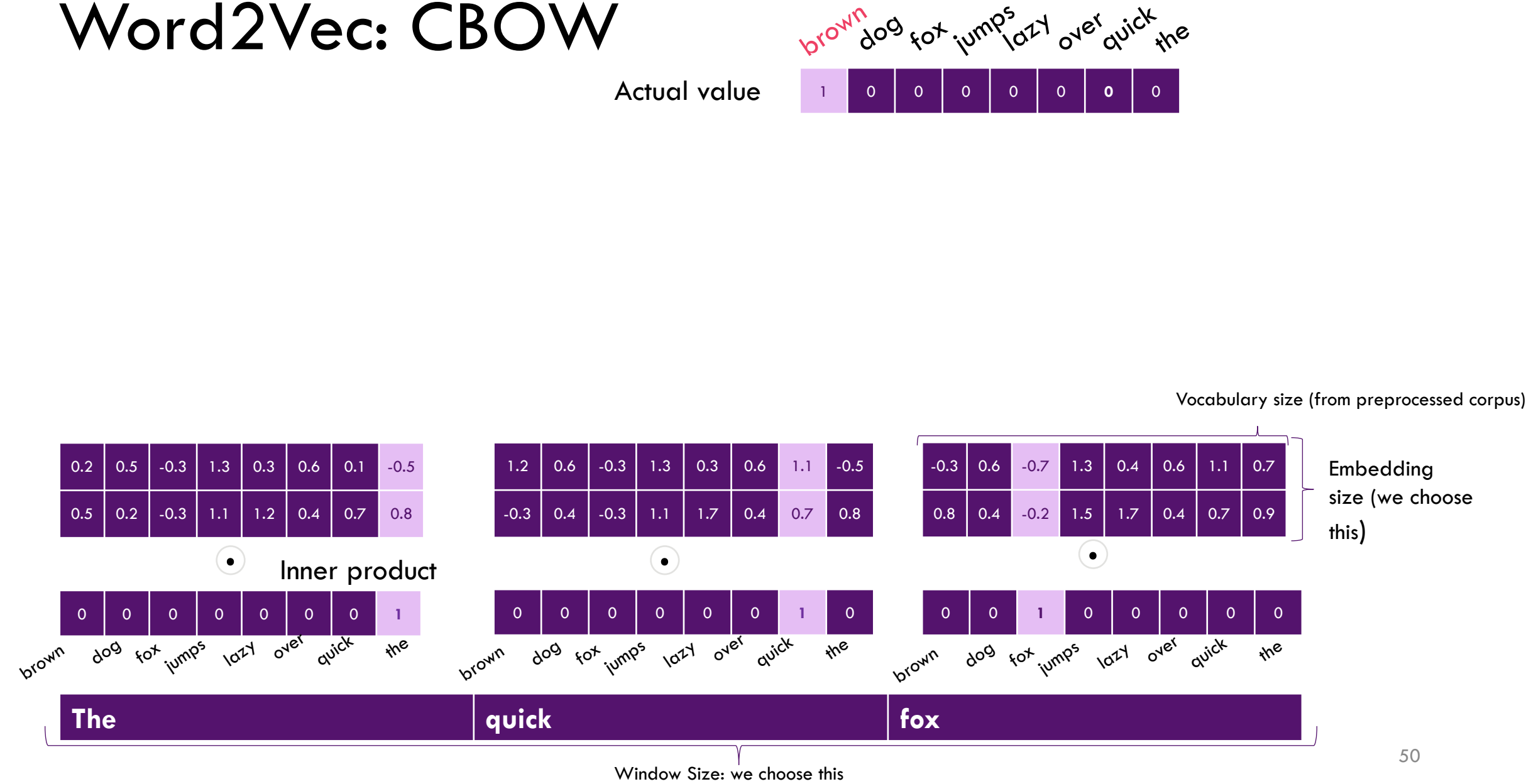
$\hat{y}_1 = w_7o_1 + w_8o_2$

Training sample:

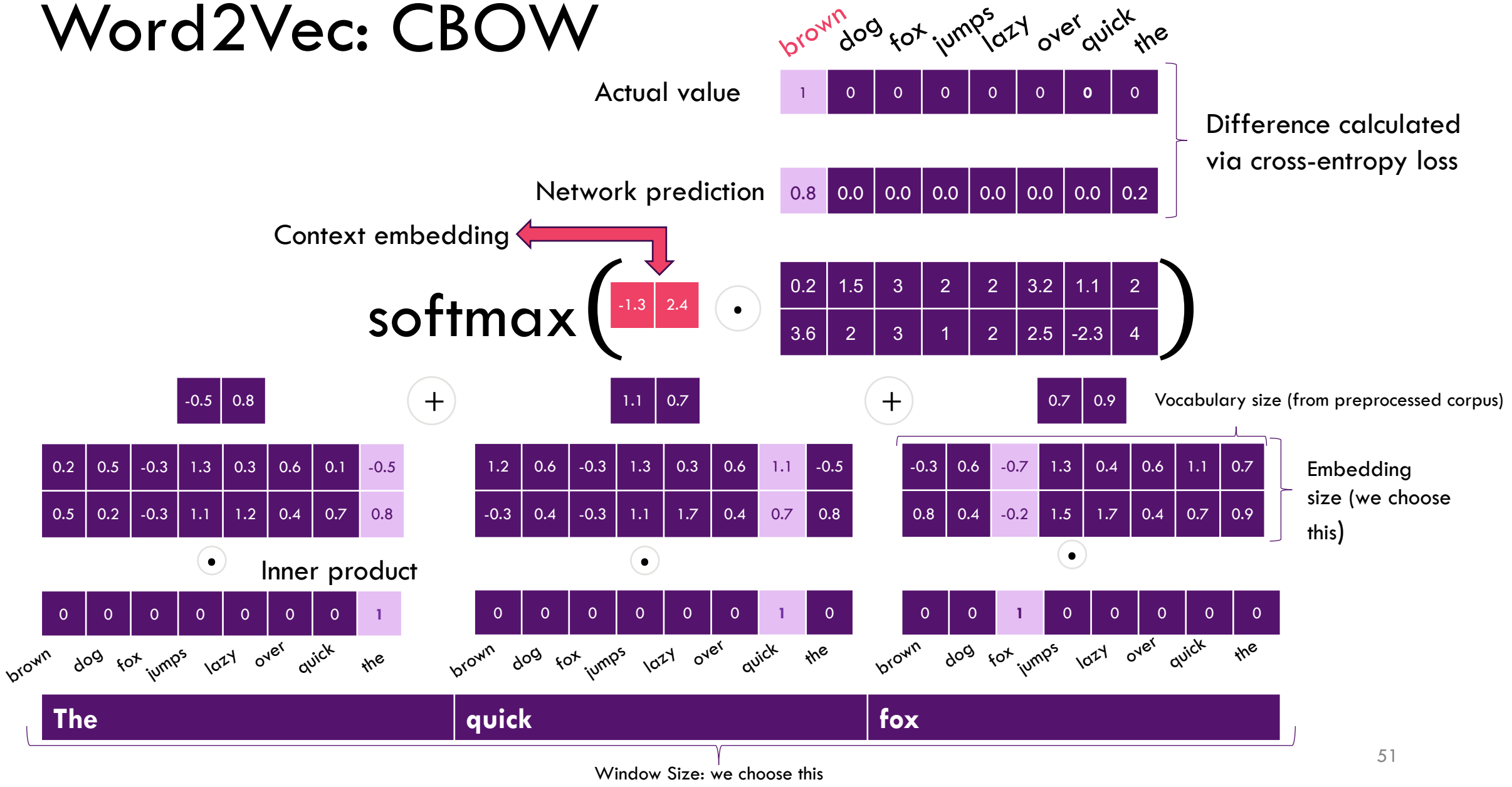
Word2Vec: CBOW



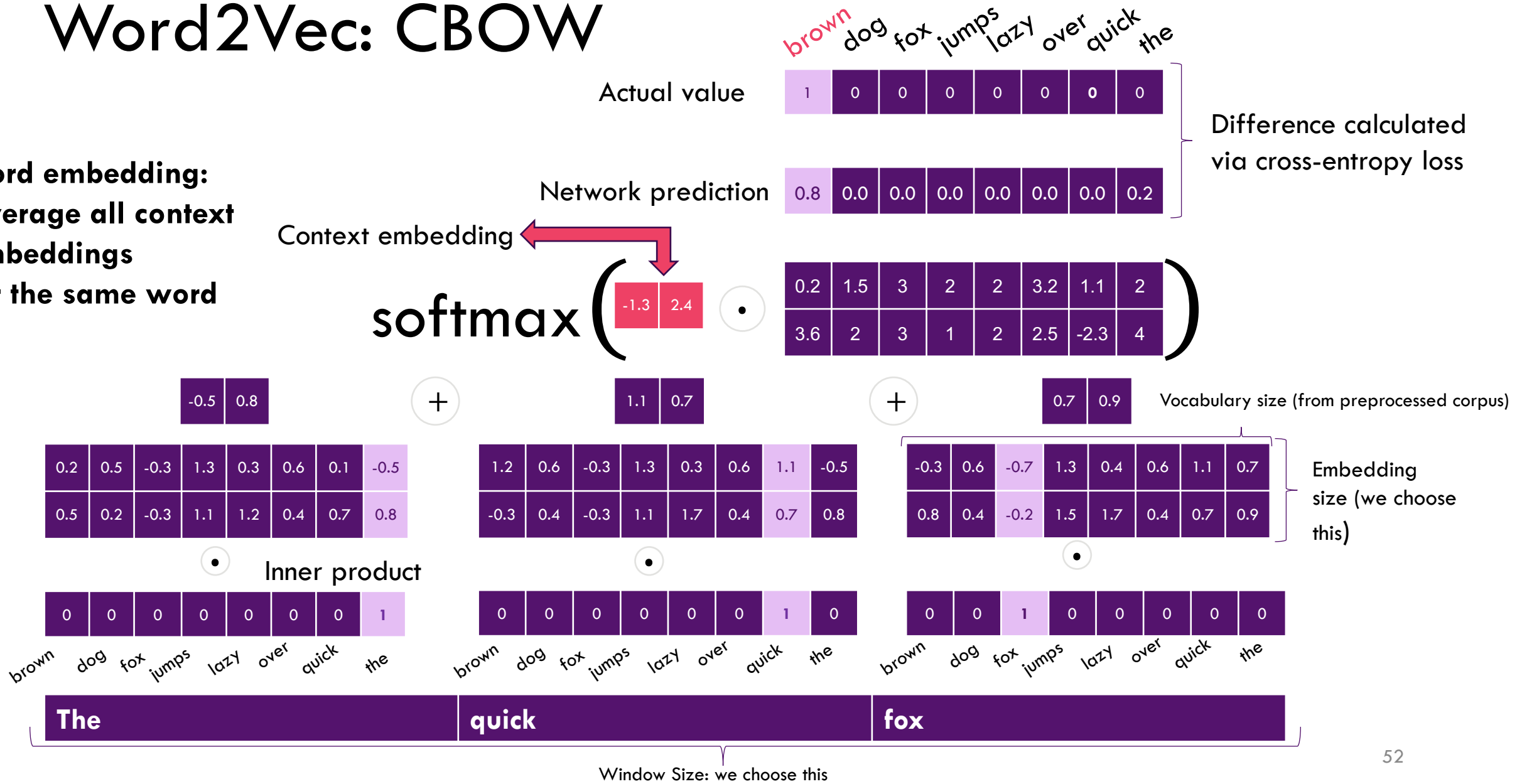
Word2Vec: CBOW



Word2Vec: CBOW



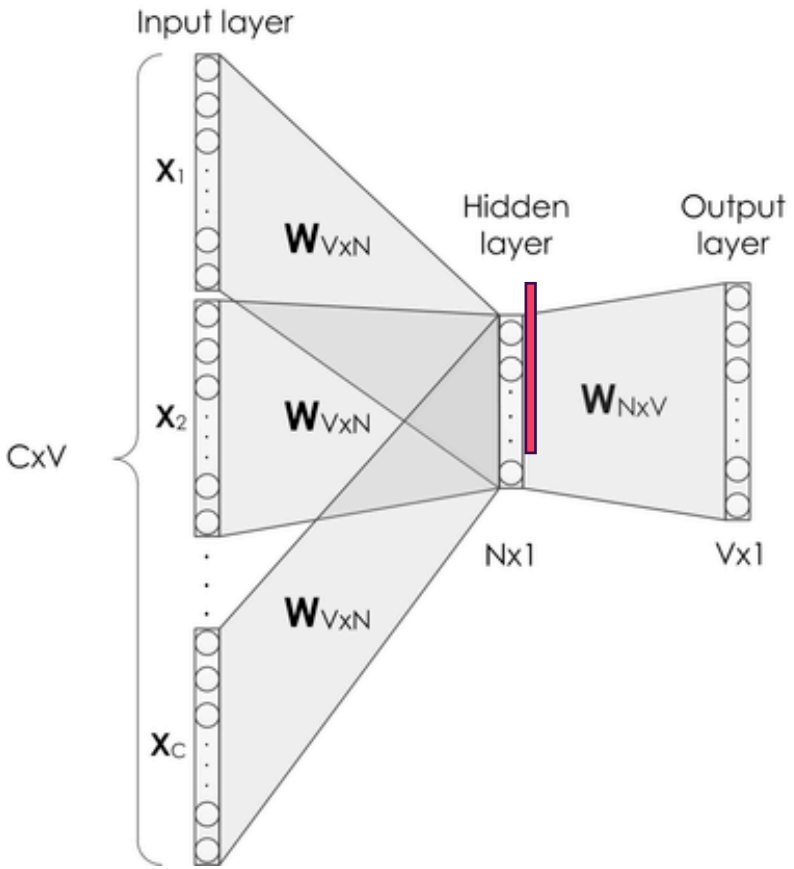
Word2Vec: CBOW



CBOW and Skip-gram Word2Vec

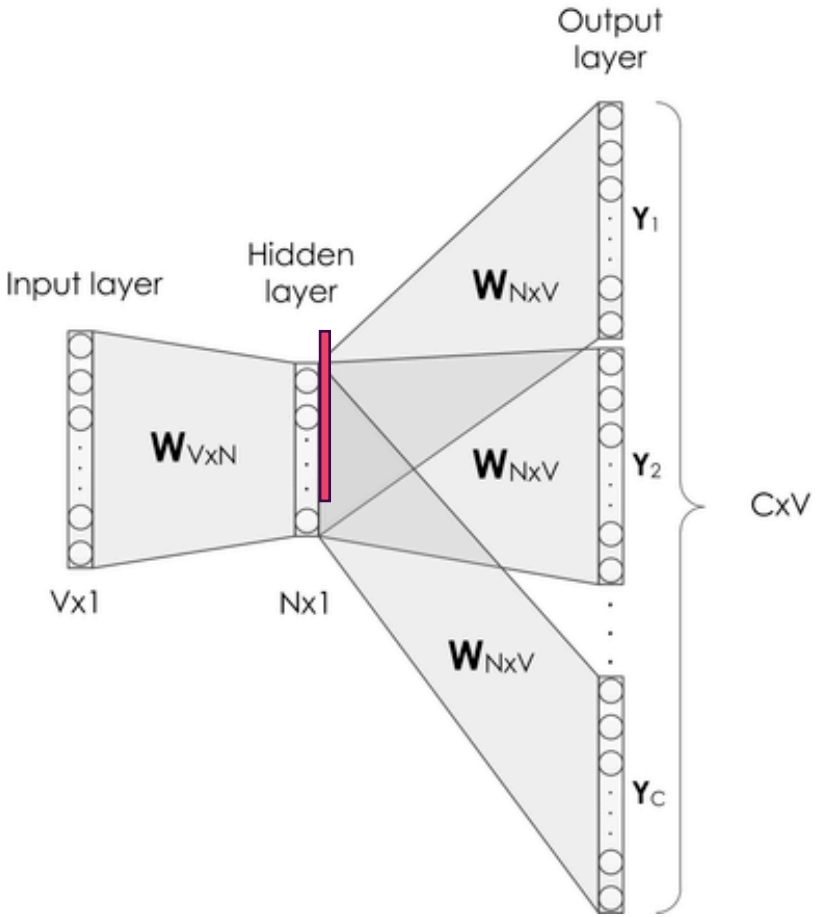
CBOW

Use context to find missing word



Skip-gram

Use missing word to find context



Tutubalina, E., & Nikolenko, S. (2017). Demographic prediction based on user reviews about medications. Computación y Sistemas, 21(2), 227-241.
https://pdfs.semanticscholar.org/15e4/98d316284bd49545f702abe9be2b1b706137.pdf

W2V Parameters

- Training type: Skipgrams vs CBOW
- Window size: How big is the context considered. Larger contexts may be beneficial but also introduce noise.
- Embedding size: Number of dimensions of embedding size. Usually the higher the better.
- Neural net parameters like learning rate, epochs etc. Often we use default.

Analogy Tasks

- Word2Vec can be used in many tasks, but it became very famous for its accuracy in analogy tasks:

$$King + Queen = Man - ?$$
$$? = King + Queen - Man$$

Replacing words with their embeddings and doing a nearest neighbor search on the result, yields:

$$Woman$$

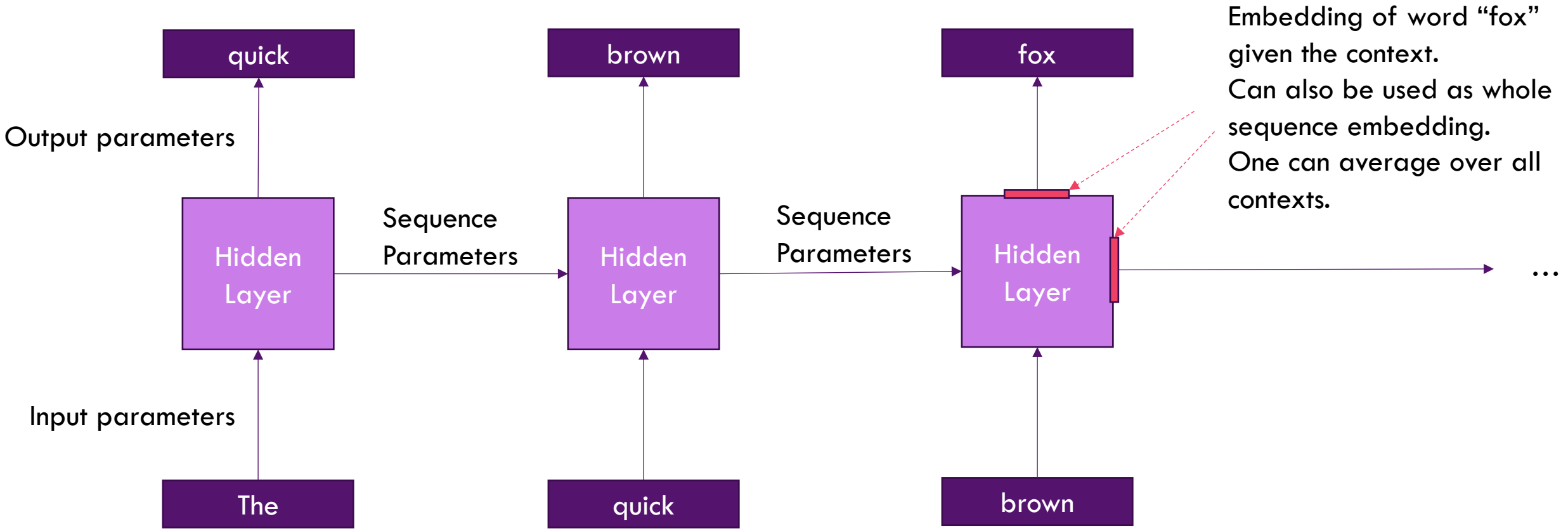
GloVe

- Designed to include global and local context.
- Relies on matrix factorization of the co-occurrence matrix (not exactly a neural net).
- Trains faster than word2vec.
- Is designed to include global properties and treat stopwords as well.
- Studies conclude that with proper parametrization, both models have similar capabilities.
- Parameters: Embedding size, window size when calculating co-occurrence matrix, learning rate etc.

Advanced Neural Models

- Such models rely on more complex neural architectures.
- They can capture contextual as well as topological and compositional relationships between words.
- Can be used to find context in arbitrary length of sequence.
- Usually can tackle more complex tasks than analogy.
- Input can be anything, e.g. from binary representations to word2vec features.
- Many hyperparameters too chose.
- Can overfit easily.

Sequence Embeddings



Sequence Embeddings II

- Very expensive models (usually train in days).
- Stateful models. Meaning that we can train over long sequences, as the model can remember the state of previous input sequences.
- Bidirectional Sequence models: What I write now is depended on what I am planning to write and what I wrote.
- Seq2Seq tasks: Translation, prediction (Language Modelling) etc.
- Compositional Embeddings: From Characters to Shakespeare…
- Context learning based on word position and not only surroundings.
- Older sequential models: Hidden Markov Models, Conditional Random fields.
- Famous NLP models: Skip-Thought Vectors, ELMo

Convolutional Embeddings (Topological)

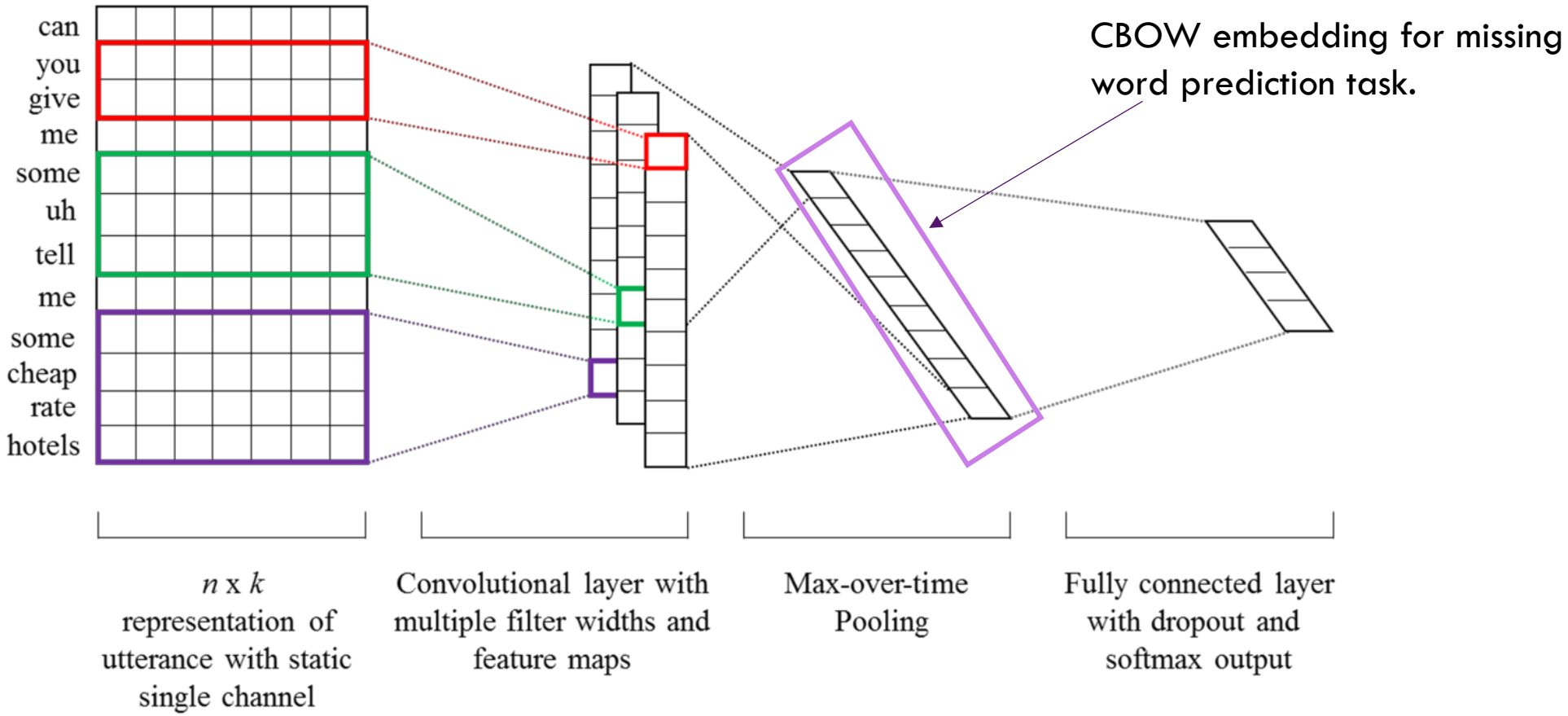


Fig. 1. CNN model architecture with single channel for an example utterance.

Xu, G., Lee, H., Koo, M. W., & Seo, J. (2017, February). Convolutional neural network using a threshold predictor for multi-label speech act classification. In 2017 IEEE international conference on big data and smart computing (BigComp) (pp. 126-130). IEEE.
https://ieeexplore.ieee.org/abstract/document/7881727/

Convolutional Embeddings II

- Cheaper and faster to train than sequential.
- Still, fixed window size.
- Non-stateful, not possible to connect large sequences.
- Incorporate topology more efficiently in learning.
- Good for sentences and small paragraphs.
- May extend to bigger texts with dictionaries.

Attention Based Embeddings

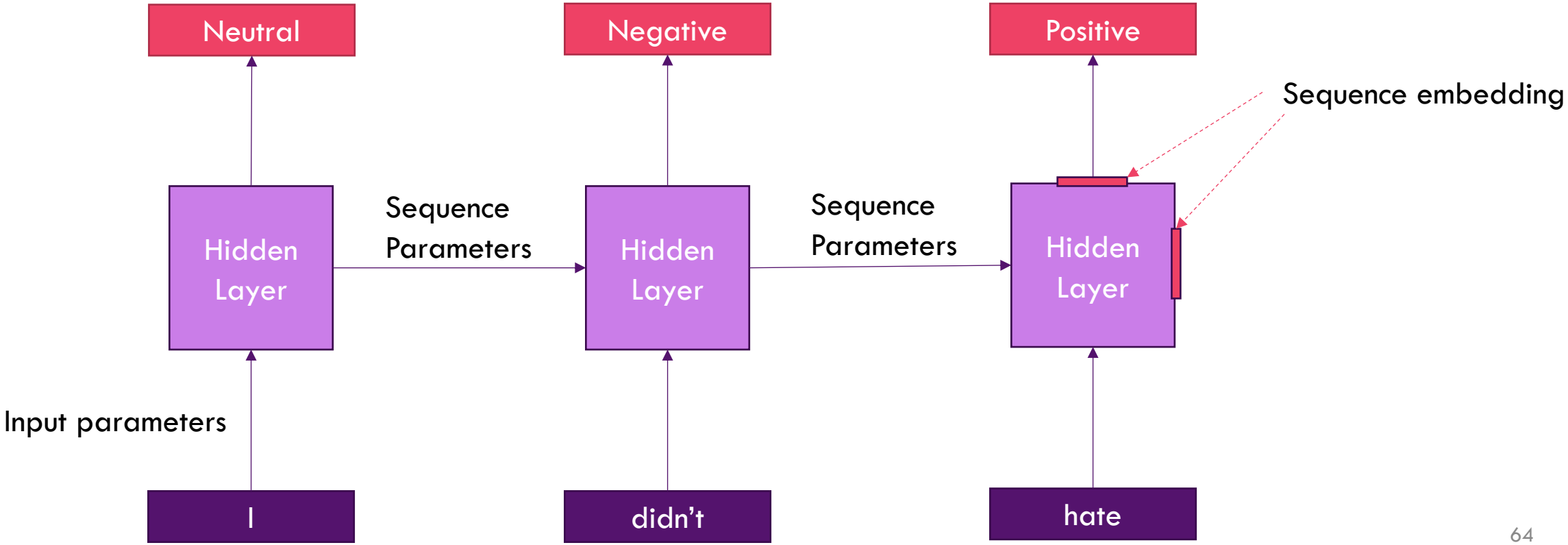
https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html

Attention Based Embeddings II

- … are along with sequence embeddings the state-of-the-art models.
- Very expensive models.
- Usually require big corpora.
- Higher performance for classification and Seq2Seq tasks.
- Can be used to produce embeddings with good transfer learning capabilities (e.g. train on one corpus and task and use on another corpus and set of tasks).

Task Specific Embeddings

- Up to now we considered methods for predicting words.
- One can train embeddings on classification tasks and then use them.



Extra Empirical Suggestions

Performance Metrics

Learning performance (how long we need to train, how good our model learns):

- Learning performance: Cross-entropy for multiclass classification.
- Binary cross-entropy: Binary classification.
- Regression: MSE.

Task performance (how well we do on the task, how good our model is):

- Expected test MSE via cross validation.
- Expected test accuracy for classification.
- Precision/Recall/ROC/AOC.

Some Practical Advice

- Regular expressions, so that you can search structure instead of only terms.
- In most languages that allow dataframe/array operations, `for`-loops are usually slower.
- Search for libraries that do low level tasks (e.g. tokenization), because usually the libraries have faster and in more correct implementations.
- Custom code mainly for uncommon tasks.
- When training machine learning models:
  i. code everything including parameter search
  ii. let them run and do other stuff,
  iii. don't check too often (e.g. check them once a day),
  iv. if no model works, start from checking the connection of your research goals to NLP tasks and available data.

Dimensionality Reduction

Most of the times, embeddings are of high dimensions. To get meaningful visualizations, we usually need to reduce to less than 3 dimensions.

Some common methods to do so are:

- **Principal Component Analysis (PCA):** Deterministic output, interpretable, not-only visualization, assumes linear relation between dimensions, focuses on global structure, can predict new data (no new fitting required).
- **t-distributed Stochastic Neighbor Embedding (TSNE):** Random output, non-interpretable, captures non-linear relationships across dimensions, finds local structures, developed for visualization, heuristic, new data → new training.
- **Uniform Manifold Approximation and Projection for Dimension Reduction (UMAP):** Random output, faster than TSNE, also captures non-linear relationships, also captures global & local structures, developed for general dimensionality reduction (you can use it for clustering), non-interpretable, theoretical guarantees, can predict new data.

https://stats.stackexchange.com/questions/238538/are-there-cases-where-pca-is-more-suitable-than-t-sne/249520#249520
https://medium.com/@dan.allison/dimensionality-reduction-with-umap-b081837354dd

Using Distances

Appropriate distances, useful for nearest neighbor, clustering etc.:

- Continuous data with many dimensions. Distance on each dimension treated equally: Euclidean.
- Higher dimensions: Fractional p-norm with $p < 1$ (may not work).
- Distributions: Kullback–Leibler divergence, Jensen-Shannon divergence.
- Nominal Data: ROCK Clustering.
- Mixed Data: Gower Distance.

How To Not Overfit

A common problem especially in neural models:

- Use more training data.
- Consider early stopping the model. More training time (e.g. epochs), higher change of overfit.
- Cross validate, usually with a big validation set in neural networks.
- Use complex models only for difficult tasks.
- Training on low frequency words usually has an overfit potential.
- Imbalanced classes in classification tasks.

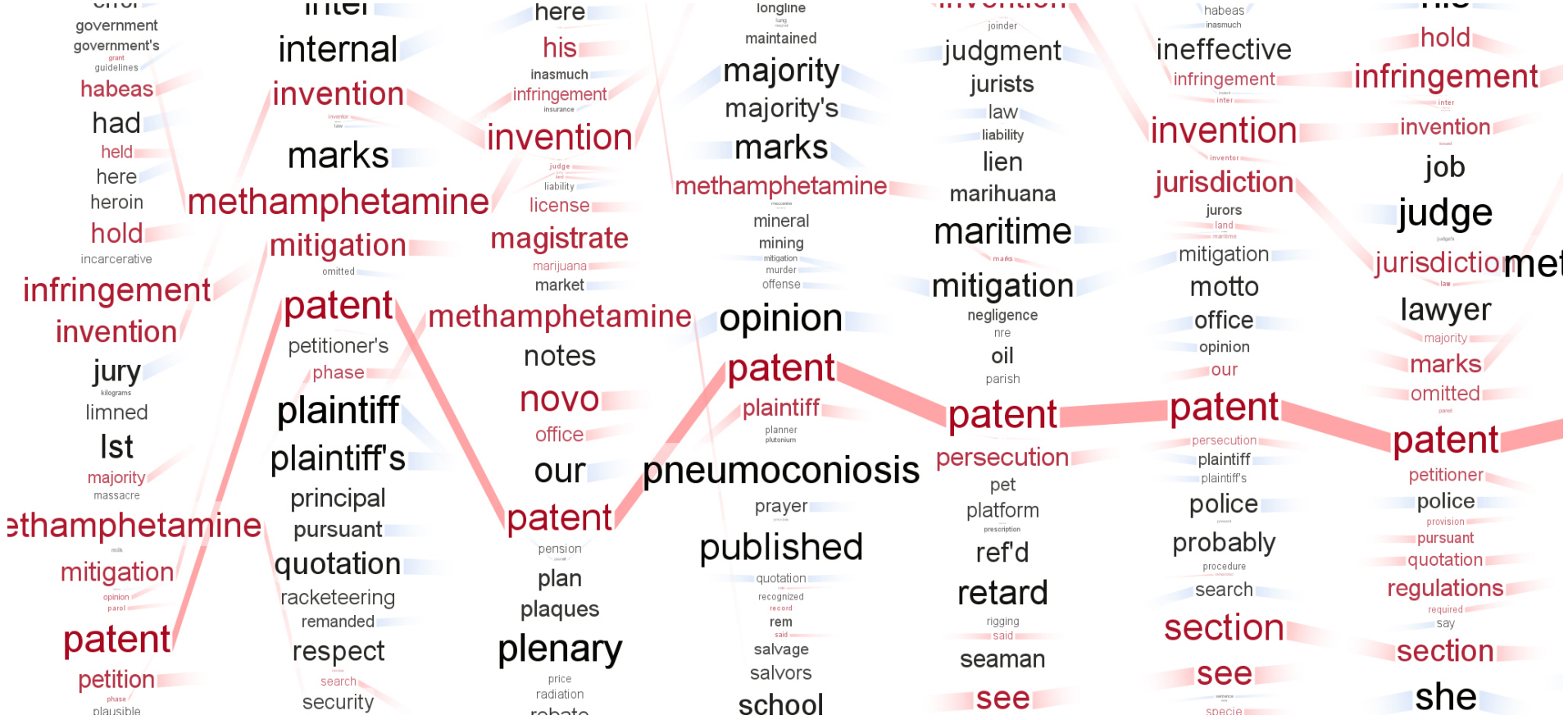
Resource Considerations (Rule of Thumb)

Characteristics of Different Methods

Method Family	Similarity/Distance	Task Specific Embeddings	Scalability	Interpretability
tf/idf	Euclidian		Low	High
LDA	Jaccard, KL		Low	High
LSH	Jaccard, extended to many		Extremely good	Low
Word2Vec, GloVe	Euclidian, Cosine		medium	Medium
Convolutional	Any	✓	good	Low
Sequential	Any	✓	medium	Low
Transformers	Any	✓	medium	Low

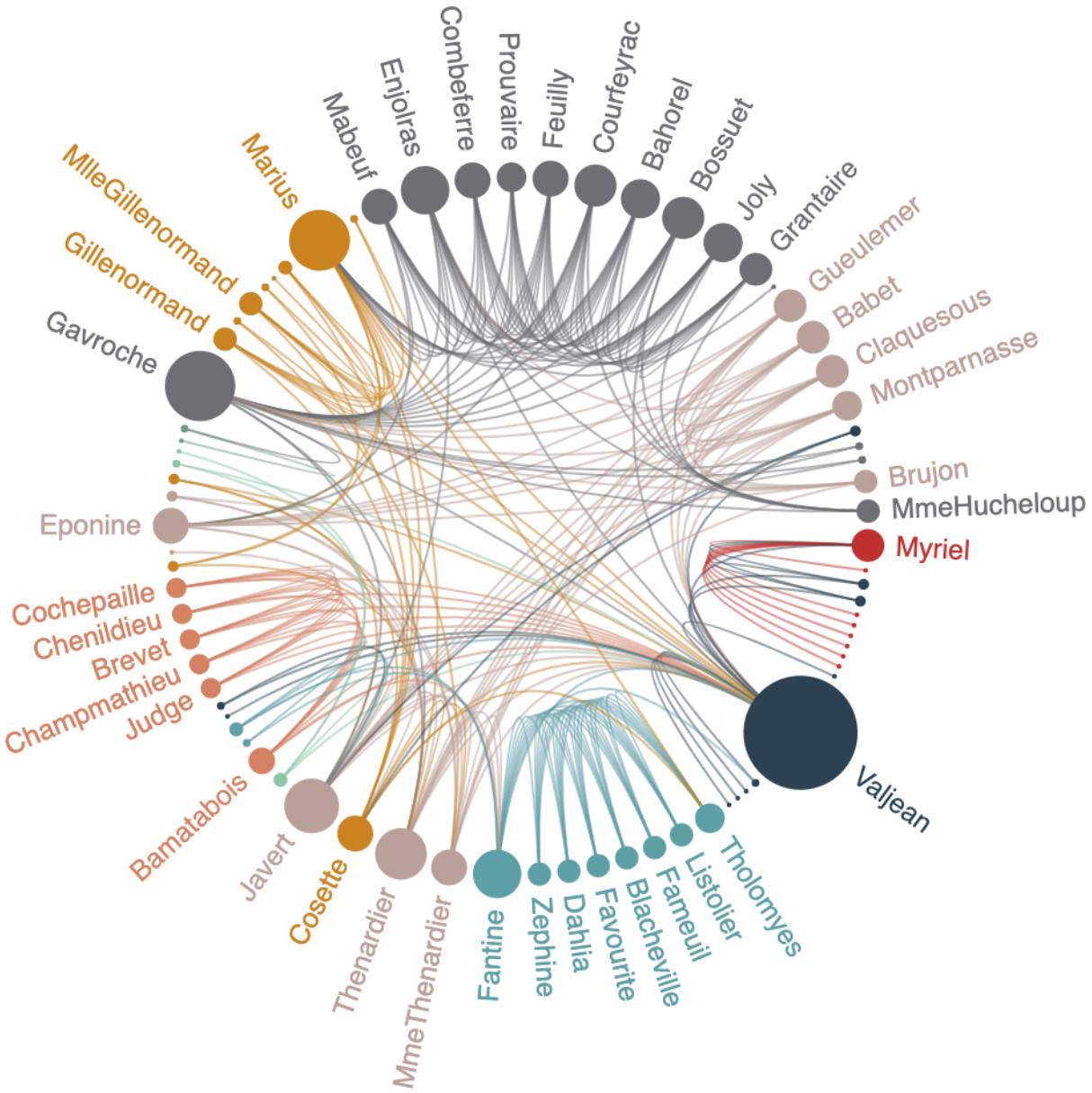
Cool Text Visualizations: WordCloud

Parallel Tag Clouds



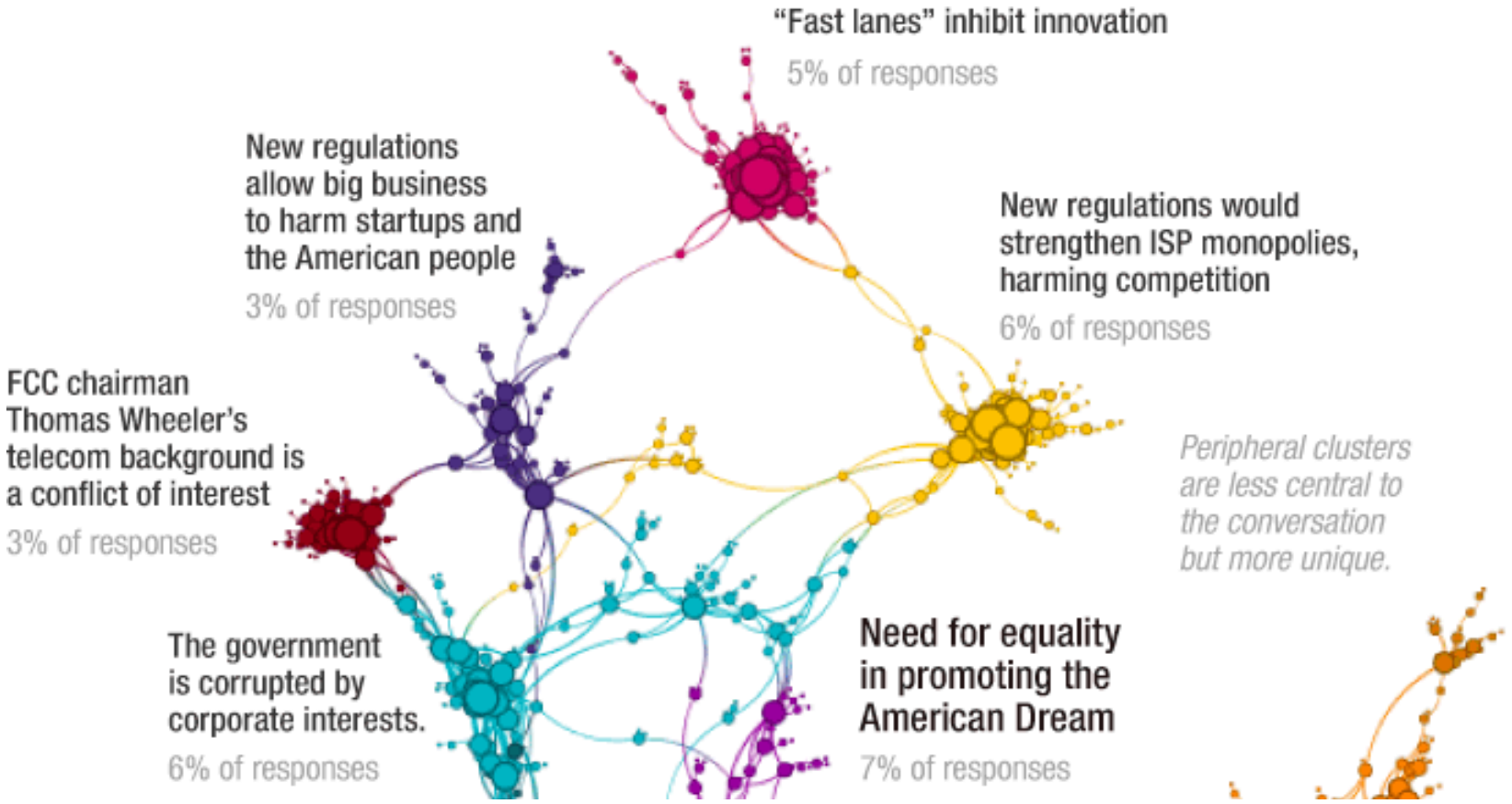
Collins, C., Viegas, F. B., & Wattenberg, M. (2009, October). Parallel tag clouds to explore and analyze faceted text corpora. In 2009 IEEE Symposium on Visual Analytics Science and Technology (pp. 91-98). IEEE.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.225.6224&rep=rep1&type=pdf

Graph Plots



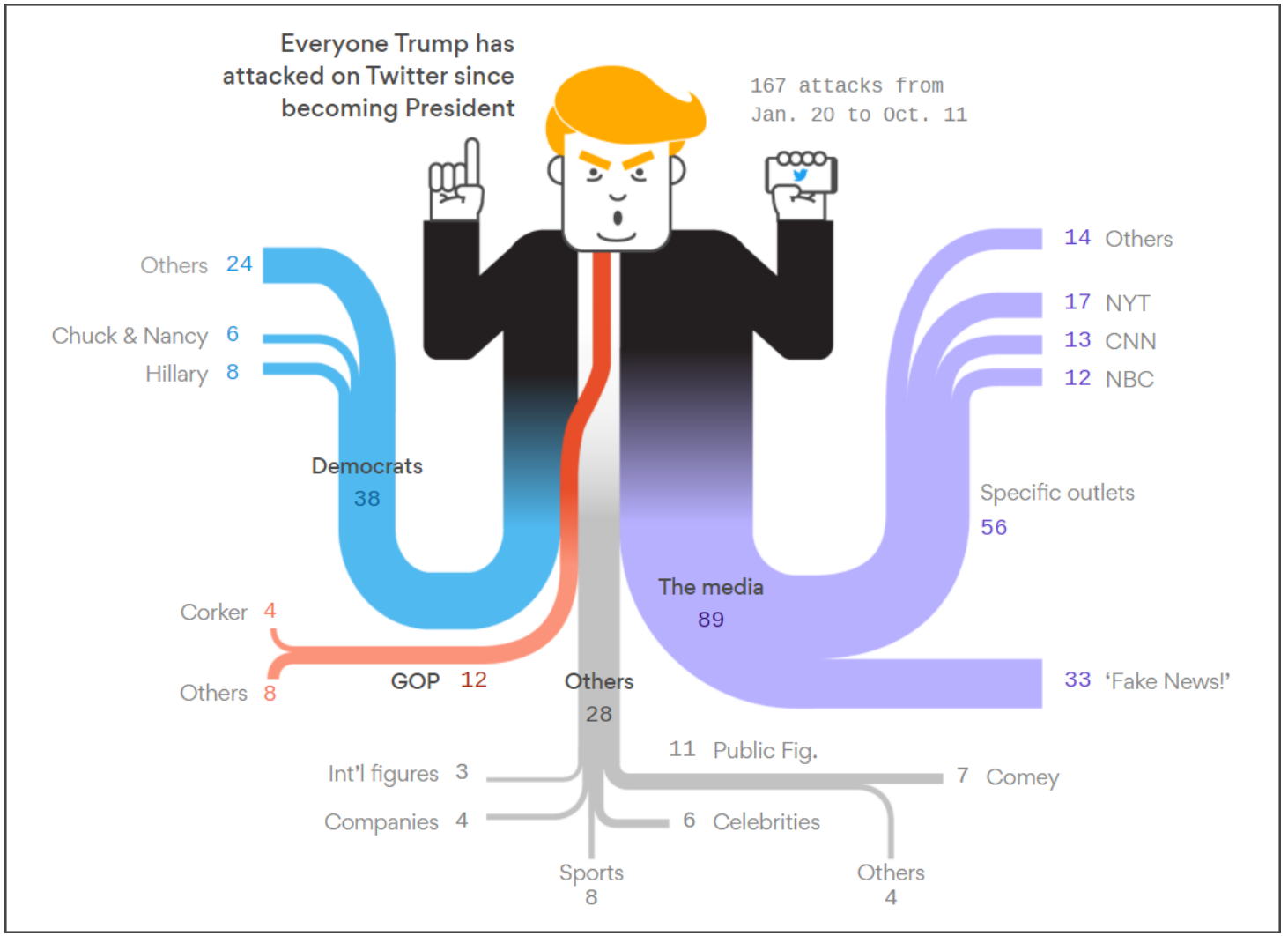
https://www.echartsjs.com/examples/en/editor.html?c=graph-circular-layout

Cluster Diagrams



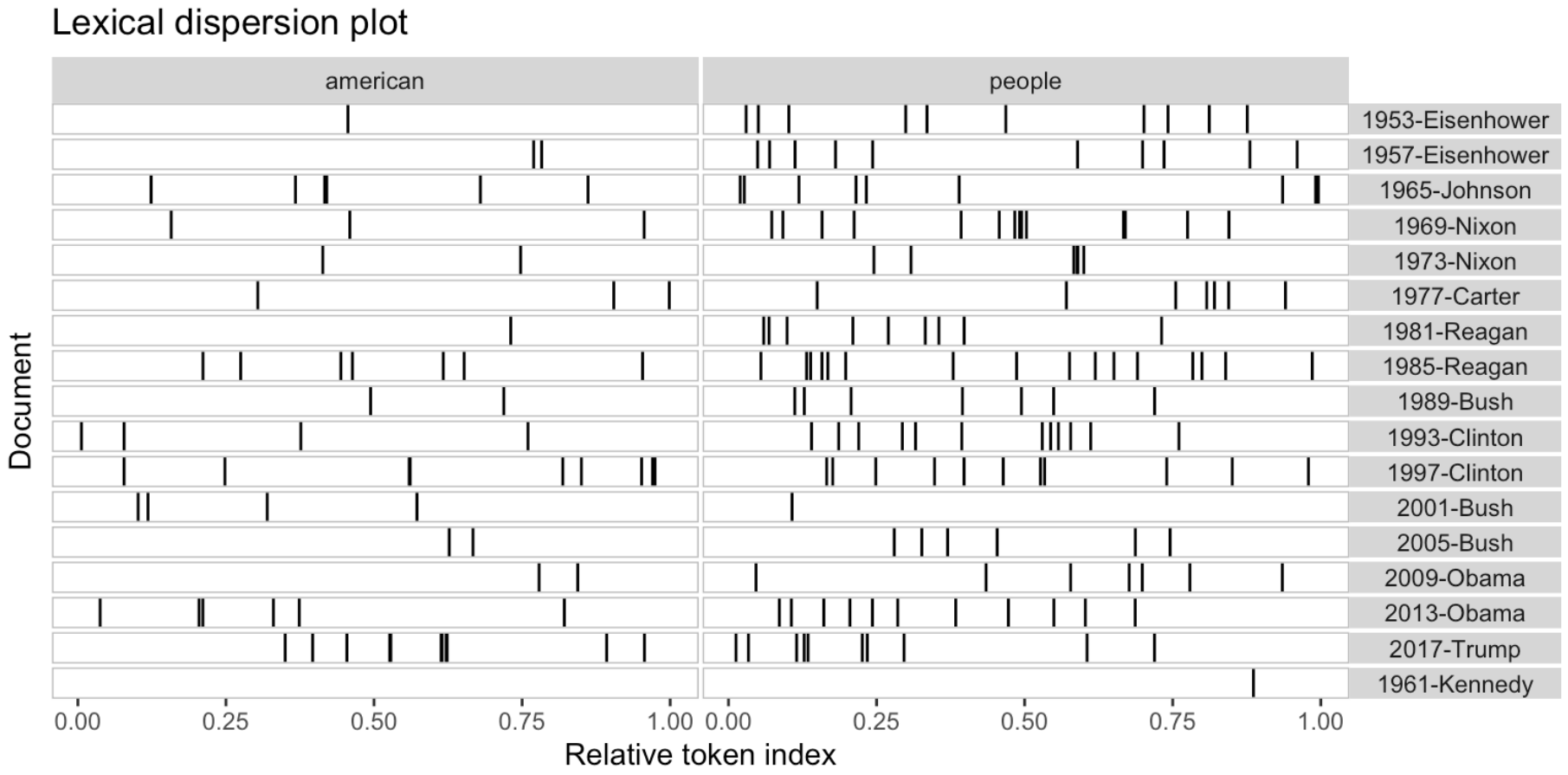
https://towardsdatascience.com/can-we-please-stop-using-word-clouds-eca2bbda7b9d

Visualizing Contextual Flows: Sankey



https://datavizblog.com/2017/10/16/sankey-diagram-who-does-president-trump-attach-the-most-on-twitter/
https://archive.nytimes.com/www.nytimes.com/interactive/2012/10/15/us/politics/swing-history.html

Lexical Dispersion Plots

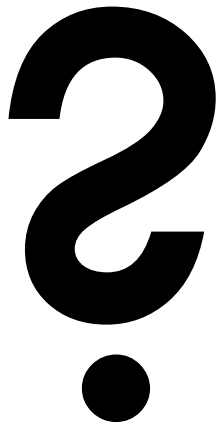


https://quanteda.io/articles/pkgdown/examples/plotting.html

Conclusions

- Important: Research goal → NLP tasks → data → model.
- Many different representations can be generated and combined.
- Different tasks → different representations.
- Different corpora → one's representations may or may not be useful for tasks on the other.
- State-of-the-art method → often works better but also needs more resources and data.
- Visualization: Don't just stick to scatter/bar plots and wordclouds.

Questions



Resources

Relevant Books

Books:

- Bender, E. M. (2013). Linguistic fundamentals for natural language processing: 100 essentials from morphology and syntax. *Synthesis lectures on human language technologies*, *6*(3), 1-184. https://www.morganclaypool.com/doi/abs/10.2200/S00493ED1V01Y201303HLT020
- Parsing, C. (2009). Speech and language processing. https://web.stanford.edu/~jurafsky/slp3/edbook_oct162019.pdf
- Manning, C. D., Manning, C. D., & Schütze, H. (1999). *Foundations of statistical natural language processing*. MIT press. https://nlp.stanford.edu/fsnlp/

Related Papers (General)

Representation Learning

- **Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828.**
  https://arxiv.org/pdf/1206.5538.pdf

Distances::

- Francois, D., Wertz, V., & Verleysen, M. (2007). The concentration of fractional distances. IEEE Transactions on Knowledge and Data Engineering, 19(7), 873-886.
  https://ieeexplore.ieee.org/abstract/document/4216305/
- Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transactions on Information theory, 37(1), 145-151.
  https://ieeexplore.ieee.org/abstract/document/61115/
- Guha, S., Rastogi, R., & Shim, K. (2000). ROCK: A robust clustering algorithm for categorical attributes. Information systems, 25(5), 345-366.
  http://www.facweb.iitkgp.ac.in/~shamik/autumn2012/dwdm/papers/ROCK%20A%20Robust%20Clustering%20Algorithm%20for%20Categorical%20Attributes%20(2000)guha00rock.pdf
- Gower, J. C. (1971). A general coefficient of similarity and some of its properties. Biometrics, 857-871.
  https://pdfs.semanticscholar.org/668b/f9f3932d29f316d68276958acf62256aaac3.pdf

Related Papers (General NLP)

NLP Tasks

- Hashimoto, K., Xiong, C., Tsuruoka, Y., & Socher, R. (2016). A joint many-task model: Growing a neural network for multiple nlp tasks. *arXiv preprint arXiv:1611.01587*. https://arxiv.org/pdf/1611.01587.pdf
- **Hotho, A., Nürnberger, A., & Paaß, G. (2005, May). A brief survey of text mining. In *Ldv Forum* (Vol. 20, No. 1, pp. 19-62). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.447.4161&rep=rep1&type=pdf**
- Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). A brief survey of text mining: Classification, clustering and extraction techniques. *arXiv preprint arXiv:1707.02919*. https://arxiv.org/abs/1707.02919

Preprocessing

- Haddi, E., Liu, X., & Shi, Y. (2013). The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, *17*, 26-32. https://www.sciencedirect.com/science/article/pii/S1877050913001385
- Jianqiang, Z., & Xiaolin, G. (2017). Comparison research on text pre-processing methods on twitter sentiment analysis. *IEEE Access*, *5*, 2870-2879. https://ieeexplore.ieee.org/abstract/document/7862202/
- Vijayarani, S., Ilamathi, M. J., & Nithya, M. (2015). Preprocessing techniques for text mining-an overview. *International Journal of Computer Science & Communication Networks*, *5*(1), 7-16. https://pdfs.semanticscholar.org/1fa1/1c4de09b86a05062127c68a7662e3ba53251.pdf

TF/IDF

- **Ramos, J. (2003, December). Using tf-idf to determine word relevance in document queries. In Proceedings of the first instructional conference on machine learning (Vol. 242, pp. 133-142). https://www.cs.rutgers.edu/~mlittman/courses/ml03/iCML03/papers/ramos.pdf**
- Wu, H. C., Luk, R. W. P., Wong, K. F., & Kwok, K. L. (2008). Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, *26*(3), 13. https://dl.acm.org/citation.cfm?id=1361686

Related Papers (Word Embeddings)

Word Embeddings:

- Stratos, K., Collins, M., & Hsu, D. (2015, July). Model-based word embeddings from decompositions of count matrices. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (pp. 1282-1291). https://www.aclweb.org/anthology/P15-1124/
- **Rong, X. (2014). word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*. https://arxiv.org/abs/1411.2738**
- Goldberg, Y., & Levy, O. (2014). word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*. https://arxiv.org/abs/1402.3722
- **Shi, T., & Liu, Z. (2014). Linking GloVe with word2vec. *arXiv preprint arXiv:1411.5595*. https://arxiv.org/abs/1411.5595**
- Pennington, J., Socher, R., & Manning, C. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543). https://www.aclweb.org/anthology/D14-1162/
- Levy, O., & Goldberg, Y. (2014, June). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 302-308). https://www.aclweb.org/anthology/P14-2050.pdf
- Liu, Y., Liu, Z., Chua, T. S., & Sun, M. (2015, February). Topical word embeddings. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*. https://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/viewPaper/9314
- Mandelbaum, A., & Shalev, A. (2016). Word embeddings and their use in sentence classification tasks. arXiv preprint arXiv:1610.08229. https://arxiv.org/abs/1610.08229

Document, Paragraph Embeddings:

- **Kusner, M., Sun, Y., Kolkin, N., & Weinberger, K. (2015, June). From word embeddings to document distances. In International conference on machine learning (pp. 957-966). http://proceedings.mlr.press/v37/kusnerb15.pdf**

Related Papers (Word Embeddings Evaluation)

- Schnabel, T., Labutov, I., Mimno, D., & Joachims, T. (2015). Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 298-307). https://www.aclweb.org/anthology/D15-1036
- Faruqui, M., Tsvetkov, Y., Rastogi, P., & Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*. https://arxiv.org/abs/1605.02276
- Iacobacci, I., Pilehvar, M. T., & Navigli, R. (2016, August). Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 897-907). https://www.aclweb.org/anthology/P16-1085
- Lau, J. H., & Baldwin, T. (2016). An empirical evaluation of doc2vec with practical insights into document embedding generation. *arXiv preprint arXiv:1607.05368*. https://arxiv.org/abs/1607.05368
- Ghannay, S., Favre, B., Esteve, Y., & Camelin, N. (2016, May). Word embedding evaluation and combination. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 300-305). http://www.lrec-conf.org/proceedings/lrec2016/pdf/392_Paper.pdf
- Zuccon, G., Koopman, B., Bruza, P., & Azzopardi, L. (2015, December). Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium* (p. 12). ACM. https://dl.acm.org/citation.cfm?id=2838936

Related Papers (Political Science Applications)

- Gurciullo, S., & Mikhaylov, S. J. (2017, October). Detecting policy preferences and dynamics in the un general debate with neural word embeddings. In 2017 International Conference on the Frontiers and Advances in Data Science (FADS) (pp. 74-79). IEEE. https://ieeexplore.ieee.org/abstract/document/8253197/
- Rheault, L., & Cochrane, C. (2019). Word Embeddings for the Analysis of Ideological Placement in Parliamentary Corpora. *Political Analysis*, 1-22. https://www.cambridge.org/core/journals/political-analysis/article/word-embeddings-for-the-analysis-of-ideological-placement-in-parliamentary-corpora/017F0CEA9B3DB6E1B94AC36A509A8A7B
- Spirling, A., & Rodriguez, P. (2019). *Word Embeddings: What works, what doesn't, and how to tell the difference for applied research.* Working paper. https://www.semanticscholar.org/paper/Word-Embeddings-What-works-%2C-what-doesn-%E2%80%99-t-%2C-and-%E2%88%97-Spirling-Rodr%C3%ADguez/4d5247af0cc487ad70813893ef945d46b2a34c11

Related Papers (Advanced Neural Models)

Recurrent Networks (Sequence):

- Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In *Advances in neural information processing systems* (pp. 3079-3087). http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning
- Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., & Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems* (pp. 3294-3302). http://papers.nips.cc/paper/5950-skip-thought-vectors
- Mueller, J., & Thyagarajan, A. (2016, March). Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*. https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/viewPaper/12195

Recursive and Topological(Tree-like):

- Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, August). Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70* (pp. 1243-1252). JMLR. org. https://dl.acm.org/citation.cfm?id=3305510
- Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013, October). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642). https://www-nlp.stanford.edu/pubs/SocherEtAl_EMNLP2013.pdf

Attention Based (Graph based):

- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems* (pp. 5998-6008). http://papers.nips.cc/paper/7181-attention-is-all-you-need
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. https://arxiv.org/abs/1810.04805

Related Pages

Word Embeddings:

- https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa
- https://machinelearningmastery.com/what-are-word-embeddings/
- https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.400.1058&rep=rep1&type=pdf
- https://medium.com/huggingface/universal-word-sentence-embeddings-ce48ddc8fc3a
- https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270
- https://blogs.rstudio.com/tensorflow/posts/2017-12-22-word-embeddings-with-keras/
- https://mchromiak.github.io/articles/2017/Sep/12/Transformer-Attention-is-all-you-need/
- https://towardsdatascience.com/word-embeddings-for-sentence-classification-c8cb664c5029
- https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010
- https://towardsdatascience.com/word2vec-made-easy-139a31a4b8ae

Related Pages II

Basic Methods:

- TF/IDF http://datameetsmedia.com/bag-of-words-tf-idf-explained/
- LDA https://towardsdatascience.com/nlp-extracting-the-main-topics-from-your-dataset-using-lda-in-minutes-21486f5aa925
- LSH https://towardsdatascience.com/understanding-locality-sensitive-hashing-49f6d1f6134

Preprocessing:

- https://www.kdnuggets.com/2017/12/general-approach-preprocessing-text-data.html

NLP Tasks:

- https://www.kdnuggets.com/2018/10/main-approaches-natural-language-processing-tasks.html
- https://natural-language-understanding.fandom.com/wiki/List_of_natural_language_processing_tasks
- https://nlp.stanford.edu/~wcmac/#papers
- https://towardsdatascience.com/nlp-extracting-the-main-topics-from-your-dataset-using-lda-in-minutes-21486f5aa925

Code and Libraries

- Text preprocessing and processing: https://quanteda.io/
- LSA: https://quanteda.io/articles/pkgdown/examples/lsa.html
- LDA: https://www.tidytextmining.com/topicmodeling.html
- GloVe: http://text2vec.org/vectorization.html
- Word2Vec: https://rdrr.io/github/bmschmidt/wordVectors/
- LSH: https://rdrr.io/cran/textreuse/man/lsh.html
- Bert: https://blogs.rstudio.com/tensorflow/posts/2019-09-30-bert-r/
- Keras: https://keras.rstudio.com/

Images and Fonts

- Hieroglyphic Fonts and study: http://users.teilar.gr/~g1951d/
- https://icons8.com/icons/set/brain-chip-ai
- https://commons.wikimedia.org/wiki/File:Hieroglyphs.jpg
- https://www.needpix.com/photo/661199/text-mining-entity-extraction-entity-recognition-text-processing-document
- https://commons.wikimedia.org/wiki/File:Parse_thicket_constructed_from_parse_trees_for_sentences.jpg
- https://mchromiak.github.io/articles/2017/Sep/12/Transformer-Attention-is-all-you-need/
- https://www.flickr.com/photos/132053576@N03/18721001439

KEEP
CALM
AND
CHECK
BACKUP SLIDES

Backup Slides

Another View on Transformers with Conv-Nets

https://mchromiak.github.io/articles/2017/Sep/12/Transformer-Attention-is-all-you-need/