Chapter NLP:II

II. Text Models

- ❑ Text Preprocessing
- ❑ Text Representation
- ❑ Text Similarity
- ❑ Text Classification
- ❑ Language Modeling
- ❑ Sequence Modeling

Text Similarity

Definition

The similarity describes the (non-)difference of two texts in either form or meaning:

1. Lexical similarity describes the closeness of form.

   Language variation: color vs. colour
   Additional words: This is shit. vs. This is *the* shit.
   Spelling errors: restaurant vs. westauwang
   Similar writing: content vs. contempt

2. Semantic similarity describes the closeness of meaning.

   Synonym: content vs. satisfied

   Paraphrase: Obama visited the capital of France. vs. Barack Obama was in Paris.

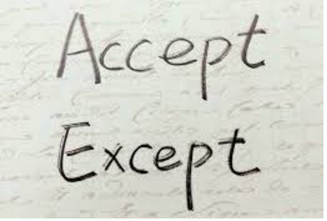
Text Similarity

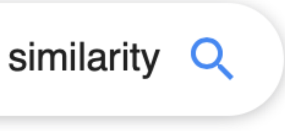
Example Applications

The precise meaning of the *difference* often depends on the application:

- Spelling correction
- Retrieval of relevant web pages
- Detection of related documents
- Paraphrase recognition
- (Near-) Duplicate or text reuse detection
- Identification of counterarguments
- Clustering
- Evaluation of machine translation and summarization

. . . and many more




similarity

Text Similarity

Similarity Measures

Text similarity can be measured by different similarity functions:

- The level of the text. token, span, or document
- The representation of the text. string or vector
- The type of similarity. lexical or semantic

Similarity functions

String difference. For lexical similarity of strings, i.e for spelling correction.

Thesaurus. For semantic similarity between tokens, i.e. synonymy.

Vector distance. For similarity between vector representations.
Depends on level and type.

Vector sequences. For similarity between multiple (ordered) vectors.

Similarity learning. For complex similarity relationships.

Text Similarity

String Similarity: Hamming Distance

The Hamming distance measures the number of substitutions required to transform one string into another.

s_1	w	e	s	t	a	u	w	a	n	g	
s_2	r	e	s	t	a	u	r	a	n	t	
Distance	1	0	0	0	0	0	1	0	0	1	= 3

However, Hamming distance is only defined for sequences of equal length.

Text Similarity

String Similarity: Edit (Levenshtein) Distance

The edit distance measures the minimum number (or cost) of editing operations needed to transform one string to another.

- Editing operations. Insertion, deletion, substitution.
- Weighted edit distance. Different edits vary in costs.

```
I N T E * N T I O N
| | | | | | | | | |
d s s   i s
| | | | | | | | | |
* E X E C U T I O N
```

How to compute edit distance?

- Sequence alignment using dynamic programming.
- Equals shortest path search in a weighted graph.

	E	X	E
I	s(I,E)	i(*,X)	
N	d(N,*)	s(N,X)	
T			

Edit distance is frequently used in Spelling correction (e.g., web search queries) or for alignment in computational biology (kind of a language problem).

“westauwang” → Did you mean “restaurant”?

Text Similarity

Resource-based Similarity: Thesaurus Lookup

What are synonyms?

- Words (or terms) that have the same meaning in some or all contexts.

  “couch” vs. “sofa” “big” vs. “large” “water” vs. “$\mathrm{H}_2$0” “vomit” vs. “throw up”

- There are hardly any perfectly synonymous terms.

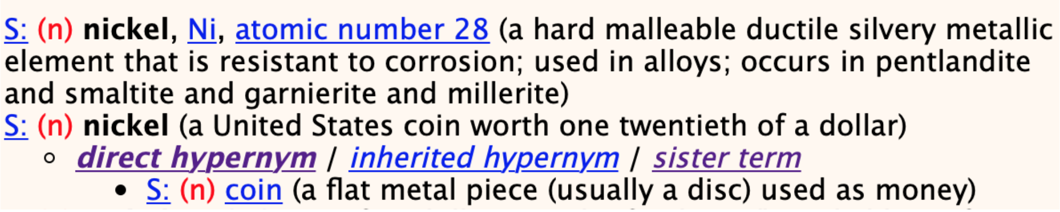
  Even seemingly identical terms usually differ in terms of politeness, slang, genre, etc.

- Synonymy is a relation between senses rather than words.

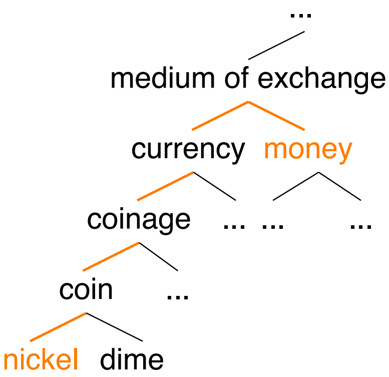
  “big” vs. “large” → “Max became kind of a <insert> brother to Linda.”

How to identify related senses?

- Compute distance in thesauri, such as *WordNet*.
  wordnetweb.princeton.edu/perl/webwn



S: (n) **nickel**, Ni, atomic number 28 (a hard malleable ductile silvery metallic element that is resistant to corrosion; used in alloys; occurs in pentlandite and smaltite and garnierite and millerite)
S: (n) **nickel** (a United States coin worth one twentieth of a dollar)
  ○ ***direct hypernym*** / *inherited hypernym* / *sister term*
    • S: (n) coin (a flat metal piece (usually a disc) used as money)



- Several libraries for such measures freely available.

Remarks:

Limitations of resource-based methods

- Many words are missing as well as basically all phrases, and also some sense connections.
- Verbs and adjectives are not as hierarchically structured as nouns.
- Thesauri are not available for all languages.

Text Similarity

Vector Similarity

- Given a collection of input texts or text spans, the goal is to compare any two instances o_1, o_2 from them.
- Comparison is done on feature-based representations (i.e., o_1 and o_2 are mapped to feature vectors $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$, respectively).

Measuring similarity between vectors

- Compare two vectors of the same representation with each other.

  $(1.0, 0.0, 0.3)$ vs. $(0.0, 0.0, 0.7)$ for $\mathbf{x} =$ (red, green, blue)

- The difference of each vector dimension is computed individually.

  1.0 vs. 0.0 0.0 vs. 0.0 0.3 vs. 0.7

- The similarity results from an aggregation of all differences.

  For example: $\frac{1.0+0.0+0.4}{3} \approx 0.467$

Remarks:

- A feature vector is an ordered set of values of the form $\mathbf{x} = (x_1, \ldots, x_m)$, where each feature x_i denotes a measurable property of an input.

  We consider only real-valued features here.

- Each instance o_j is mapped to a vector $\mathbf{x}^{(j)} = (x_1^{(j)}, \ldots, x_m^{(j)})$ where $x_i^{(j)}$ denotes the value of feature x_i for o_j.

  We consider only values normalized to the range $[0, 1]$ here.

- Since Distance can be seen as the inverse of similarity, similarity functions form the basis for many clustering algorithms. Clustering mostly relies on vector-based similarity measures.

- Numerous vector-based measures are found in the literature [Cha, 2007].

Text Similarity

Vector Similarity: Distance Functions

Properties of a distance function (aka metric)

- Non-negativity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;\geq\; 0$
- Identity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(1)}) \;=\; 0$
- Symmetry. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; d(\mathbf{x}^{(2)}, \mathbf{x}^{(1)})$
- Subadditivity. $d(\mathbf{x}^{(1)}, \mathbf{x}^{(3)}) \;\leq\; d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) + d(\mathbf{x}^{(2)}, \mathbf{x}^{(3)})$

  Clustering actually does not necessarily require subadditivity.

Distance computation in clustering

- Internally, clustering algorithms compute distances between instances.

	x_1	x_2	$\ldots$	x_m
$\mathbf{x}^{(1)}$	$x_1^{(1)}$	$x_2^{(1)}$	$\ldots$	$x_m^{(1)}$
$\mathbf{x}^{(2)}$	$x_1^{(2)}$	$x_2^{(2)}$	$\ldots$	$x_m^{(2)}$
$\vdots$				
$\mathbf{x}^{(n)}$	$x_1^{(n)}$	$x_2^{(n)}$	$\ldots$	$x_m^{(n)}$

	$\mathbf{x}^{(1)}$	$\mathbf{x}^{(2)}$	$\ldots$	$\mathbf{x}^{(n)}$
$\mathbf{x}^{(1)}$	0	$d(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})$	$\ldots$	$d(\mathbf{x}^{(1)}, \mathbf{x}^{(n)})$
$\mathbf{x}^{(2)}$	-	0	$\ldots$	$d(\mathbf{x}^{(2)}, \mathbf{x}^{(n)})$
$\vdots$				
$\mathbf{x}^{(n)}$	-	-	$\ldots$	0

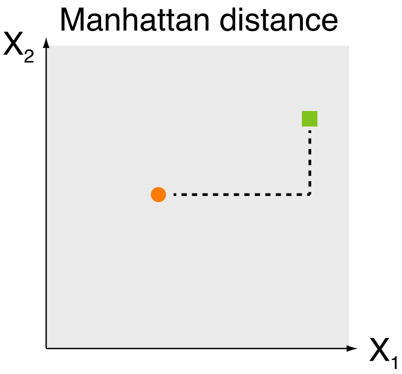
Text Similarity

Vector Similarity: Manhattan Similarity

Manhattan distance (aka city block distance)

- The Manhattan distance is the sum of all absolute differences between two feature vectors.

$$\mathit{dist}_{\mathit{Manhattan}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \sum_{i=1}^{m} |x_i^{(1)} - x_i^{(2)}|$$



Manhattan similarity

- If all feature values are normalized to $[0, 1]$, the Manhattan similarity is:

$$\mathit{sim}_{\mathit{Manhattan}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = 1 - \frac{\mathit{dist}_{\mathit{Manhattan}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})}{m}$$

Notice

- Manhattan distance and Euclidean distance are both special cases of the *Minkowski distance*.

$$\mathit{dist}_{\mathit{Minkowski}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \sqrt[p]{\sum_{i=1}^{m} |\mathbf{x}_i^{(1)} - \mathbf{x}_i^{(2)}|^p} \quad \text{for any } p \in \mathbb{N}^+$$

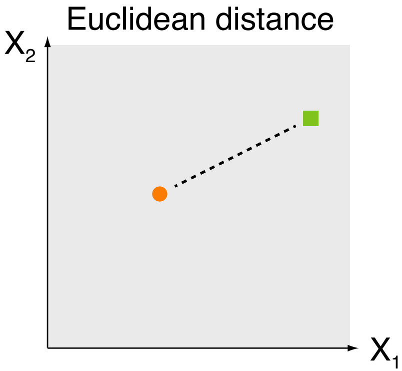
Text Similarity

Vector Similarity: Euclidean Similarity

Euclidean distance

- The Euclidean distance captures the absolute straight-line distance between two feature vectors.

$$\mathit{dist}_{\mathit{Euclidean}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; \sqrt{\sum_{i=1}^{m} |x_i^{(1)} - x_i^{(2)}|^2}$$



Euclidean similarity

- If all feature values are normalized to $[0, 1]$, the Euclidean similarity is:

$$\mathit{sim}_{\mathit{Euclidean}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) \;=\; 1 - \frac{\mathit{dist}_{\mathit{Euclidean}}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)})}{\sqrt{m}}$$

Notice

- Euclidean spaces generalize to any number of dimensions $m \geq 1$.
- Here, this means to any number of features.

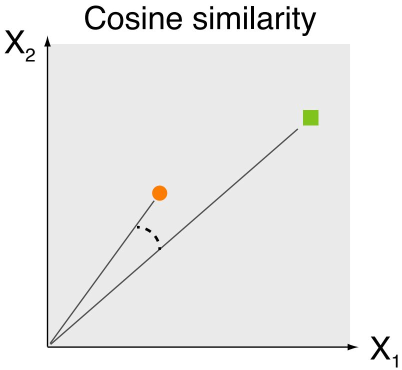
Text Similarity

Vector Similarity: Cosine Similarity

Cosine similarity (aka cosine score)

- Cosine similarity captures the cosine of the angle between two feature vectors.
- The smaller the angle, the more similar the vectors.

  This works because cosine is maximal for 0°.
- $||\mathbf{x}||$ denotes the L2 norm of vector $\mathbf{x}$:



$$sim_{Cosine}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{\mathbf{x}^{(1)} \cdot \mathbf{x}^{(2)}}{||\mathbf{x}^{(1)}|| \cdot ||\mathbf{x}^{(2)}||} = \frac{\sum_{i=1}^{m} x_i^{(1)} \cdot x_i^{(2)}}{\sqrt{\sum_{i=1}^{m} x_i^{(1)2}} \cdot \sqrt{\sum_{i=1}^{m} x_i^{(2)2}}}$$

Notice

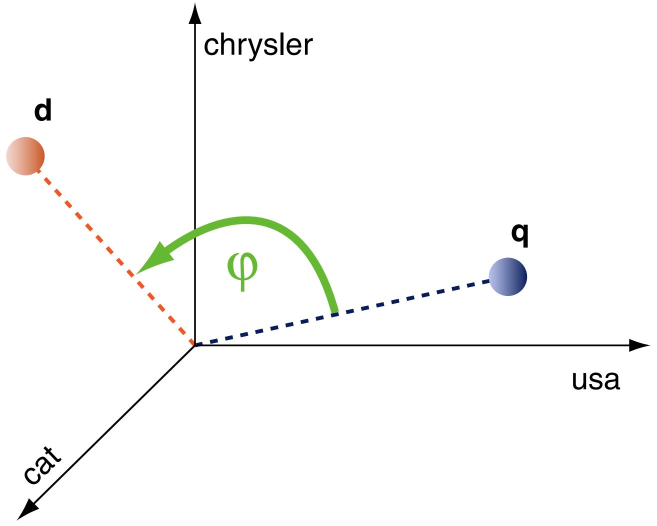
- The cosine similarity abstracts from the length of the vectors.
- Angle computation works for any number of dimensions.
- Cosine similarity is the most common similarity measure.

Text Similarity

Vector Similarity: Cosine Similarity

$$\mathbf{d} = \begin{pmatrix} \texttt{chrysler} & w_1 \\ \texttt{usa} & w_2 \\ \texttt{cat} & w_3 \\ \texttt{dog} & w_4 \\ \texttt{mouse} & w_5 \end{pmatrix} = \begin{pmatrix} \texttt{chrysler} & 1 \\ \texttt{usa} & 4 \\ \texttt{cat} & 3 \\ \texttt{dog} & 7 \\ \texttt{mouse} & 5 \end{pmatrix}$$

$$\mathbf{d_i} = \begin{pmatrix} \texttt{chrysler} & 0.1 \\ \texttt{usa} & 0.4 \\ \texttt{cat} & 0.3 \\ \texttt{dog} & 0.7 \\ \texttt{mouse} & 0.5 \end{pmatrix}, \mathbf{d_j} = \begin{pmatrix} \texttt{chrysler} & 0.2 \\ \texttt{usa} & 0.1 \\ \texttt{cat} & 0.5 \\ \texttt{ostrich} & 0.1 \\ \texttt{elephant} & 0.1 \end{pmatrix}$$



The angle φ between $\mathbf{d_i}$ and $\mathbf{d_j}$ is about $51°$, $\cos(\varphi) \approx 0.63$.

Text Similarity

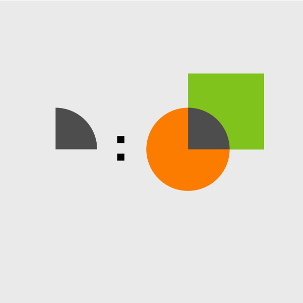
Vector Similarity: Jaccard Similarity

Jaccard similarity coefficient (aka Jaccard index)

- The Jaccard coefficient captures how large the intersection of two sets is compared to their union.
- With respect to vector representations, this makes at least sense for Boolean features.

  For others, if there is a reasonable way of thresholding.

Jaccard similarity



$$sim_{Jaccard}(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = \frac{|\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}{|\mathbf{x}^{(1)} \cup \mathbf{x}^{(2)}|} = \frac{|\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}{|\mathbf{x}^{(1)}| + |\mathbf{x}^{(2)}| - |\mathbf{x}^{(1)} \cap \mathbf{x}^{(2)}|}$$

$$= \frac{\sum_{x_i^{(1)}=x_i^{(2)}} 1}{m + m - \sum_{x_i^{(1)}=x_i^{(2)}} 1}$$

Notice

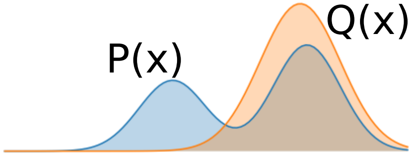
- The Jaccard similarity does *not* consider the size of the difference between feature values.

Text Similarity

Vector Similarity: Divergence

Kullback–Leibler–Divergence (KL)

- A measure of how one probability distribution is different from a second in terms of information gain (asymmetric measure, does not qualify as a statistical metric of spread - it also does not satisfy the triangle inequality)



$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log \left(\frac{P(x)}{Q(x)} \right)$$

Jenson-Shannon-Divergence (JSD)

- JSD is based on the Kullback–Leibler divergence, with some notable (and useful) differences, including that it is symmetric and it always has a finite value.

$$\begin{aligned} sim_{\mathrm{JSD}}(P(x) \parallel Q(x)) &= 1 - \left(\frac{1}{2} D_{\mathrm{KL}}(P(x) \parallel M(x)) + \frac{1}{2} D_{\mathrm{KL}}(Q(x) \parallel M(x)) \right) \\ M(x) &= \frac{1}{2}(P(x) + Q(x)) \end{aligned}$$

Notice

- This kind of distances are used in probability mathematical spaces which are not linear (e.g. Multinomial Distributions in Topic Models)

Text Similarity

Vector Similarity: When to Use What Measure?

Comparison of the measures

- ❑ Cosine similarity. Puts the focus on those properties that occur. Targets situations where a vector's direction matters rather than its length.
  A prominent use case is matching queries with documents in web search.
- ❑ Jaccard similarity. Seems less precise than cosine similarity, but this also makes it more robust (it "overfits" less).
- ❑ Euclidean and Manhattan. Target situations where a value of 0 does not mean the absence of a property.
- ❑ Euclidean or Manhattan. Depends on whether sensitivity to outliers in certain dimensions is preferred or not.
- ❑ Divergence. If the text representation is expressed in terms of probability distributions.

It is not always clear what measure will prove best. One way to deal with this is to simply evaluate different measures.

Remarks:

Limitation of vector-based measures in NLP

- Similarity is defined based on corresponding feature values $x_j^{(1)}, x_j^{(2)}$.
- Most features in NLP are derived directly from text spans.
- Similarity between different forms with similar meaning is missed . . .

  "traveling" vs. "travelling" "woodchuck" vs. "groundhog" "Trump" vs. "The President"

- . . . unless such differences are accounted for.

Vector-based methods can be used to compare both document vectors and word vectors.

- If a vector encodes semantic information, then vector-based similarity functions measure semantic similarity.
- Semantic representations of documents in singular vectors are still a topic of research.

Similarity Measures

Vector Sequence Similarity

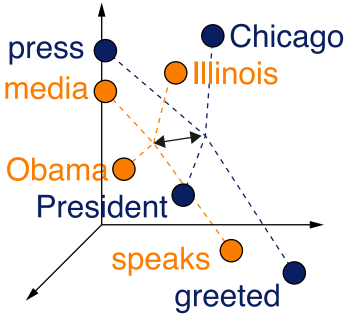
- Text can be represented as sequences of vectors (tensors) as a semantically-rich alternative to the bag-of-words model.
- Similarity can be measured on either a composit representation via Sentence Embedding or on the individual, aligned vectors via Word Mover Distance.
- Sentence embeddings can be obtained by averaging all vectors element-wise, or by training a machine learning model to predict semantic similarity.

Vector Average Similarity [Iyyer et al. 2015]

The similarity is measured on the element-wise average of all vectors in each span. Cosine is often used as the measure.

s^1: Obama speaks to the media in Illinois

s^2: The press is greeted by the President in Chicago



$$\mathit{sim}_{\mathit{Cosine}}\Big(\sum_{i=1}^{|s^1|}\frac{s_i^1}{|s^1|},\sum_{i=1}^{|s^2|}\frac{s_i^2}{|s^2|}\Big)$$

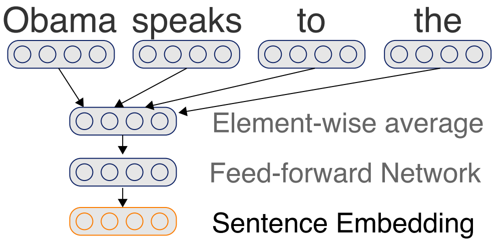
Similarity Measures

Vector Sequence Similarity: Sentence Embedding

- Text can be represented as sequences of vectors (tensors) as a semantically-rich alternative to the bag-of-words model.
- Similarity can be measured on either a composit representation via Sentence Embedding or on the individual, aligned vectors via Word Mover Distance.
- Sentence embeddings can be obtained by averaging all vectors element-wise, or by training a machine learning model to predict semantic similarity.

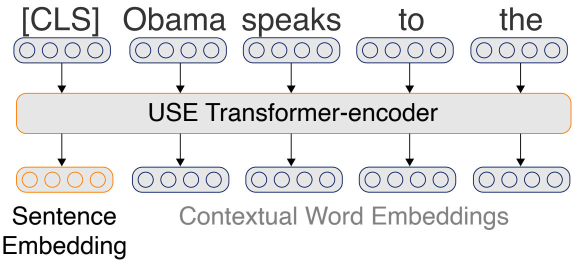
Deep Average Networks [Iyyer et al. 2015]

- Unordered composition
- Fast, good for classification



Universal Sentence Encoder [Cer et al. 2018]

- Syntactic Composition
- Very powerful, very difficult to train



Similarity Measures

Vector Sequence Similarity: Word Mover Distance

Encoding similarities in feature vectors

- String similarities can be used in diverse ways within features.

  Frequency of ~~“money”~~ the sense “the most common medium of exchange”

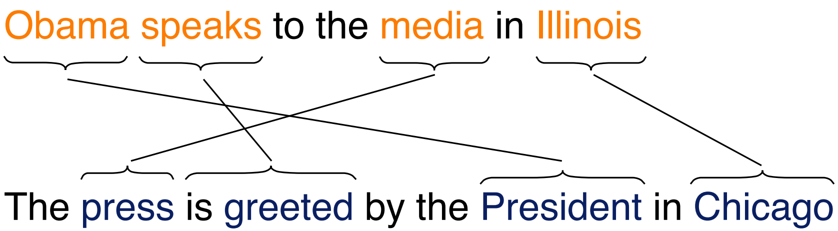
  Frequency of all writings of “traveling”

- Where reasonable, embeddings can simply be used as feature vectors.

  “nickel” → (0.14, 0.03, 0.44, . . . , 0.22)   “money” → (0.18, 0.06, 0.49, . . . , 0.01)

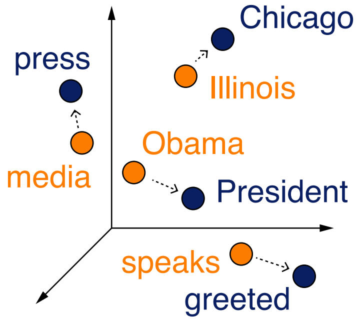
Word Mover’s Distance [Kusner et al., 2015]

- The distance of the optimal alignment of two texts.



Obama speaks to the media in Illinois

The press is greeted by the President in Chicago



- Represents texts by sequences of word embeddings.