Incorporating Embeddings in Classification Tasks

2023/04/20
Po-Ya Angela Wang
(Amber)

K means Clustering

Topic Classification

Topic Modeling

Classification tasks

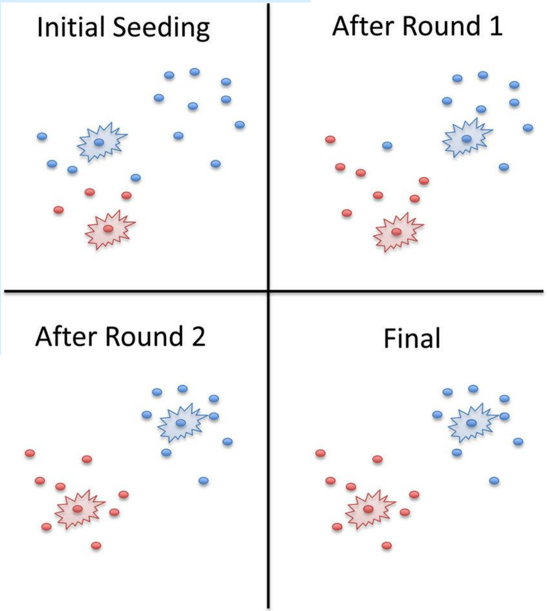
Clustering

1. classifying the given data into k clusters by defining k centroids
2. minimizing a chosen Euclidean distance between a data point and cluster center

[(PDF) BamBam: Genome sequence analysis tools for biologists]

$$\text{Euclidean distance} = \sqrt{(\text{observe value} - \text{centroid value})^2 + (\text{observe value} - \text{centroid value})^2}$$

$$\text{Euclidean distance} = \sqrt{((X_x - X_1))^2 + (X_y - Y_1)^2}$$



1. Random choice
2. Distance from each data point to the centroid
3. Closer ones belong to the same cluster
4. New centroid of each cluster (average)
5. Repeat 2-4
6. Till the the centroid value stay the same

NLP tasks

Topic classification

Topic Modeling

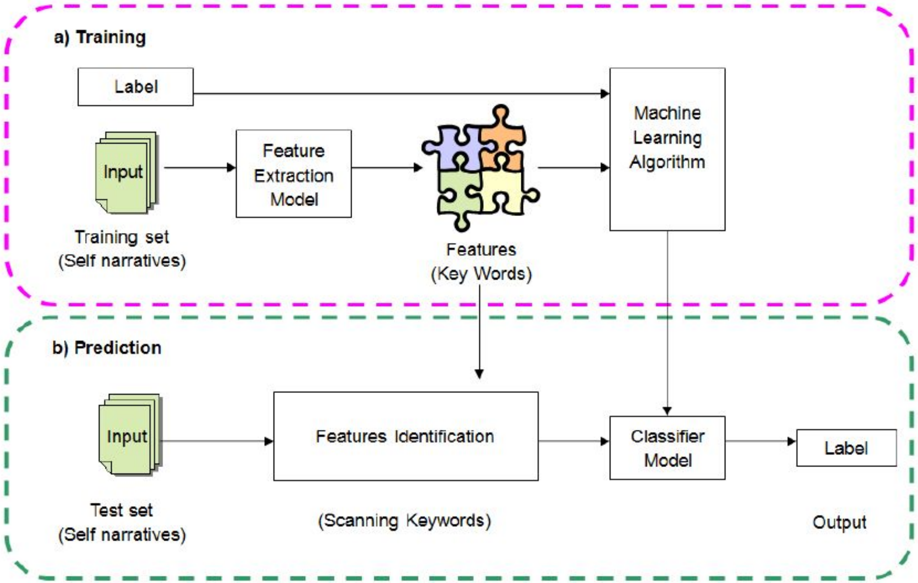
Supervised

Unsupervised

NLP tasks

Topic classification

Supervised

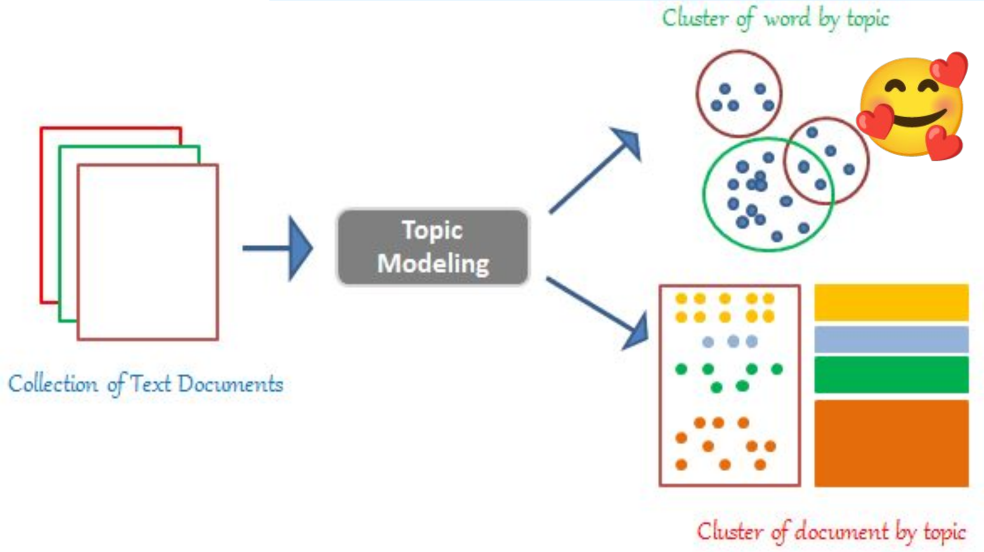


[Classifying Unstructured Textual Data Using the Product Score Model: An Alternative Text Mining Algorithm]

NLP tasks

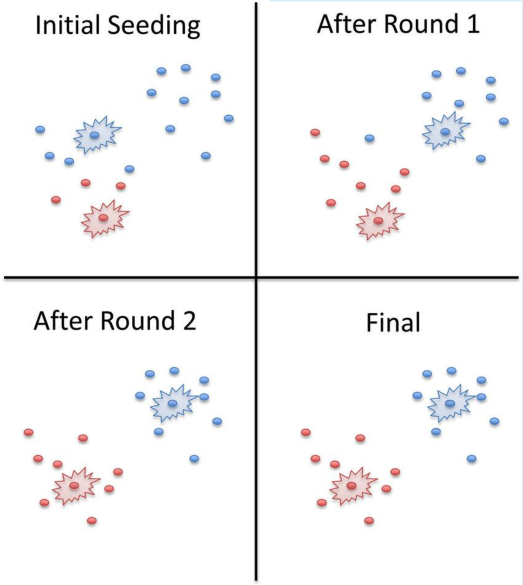
Topic Modeling

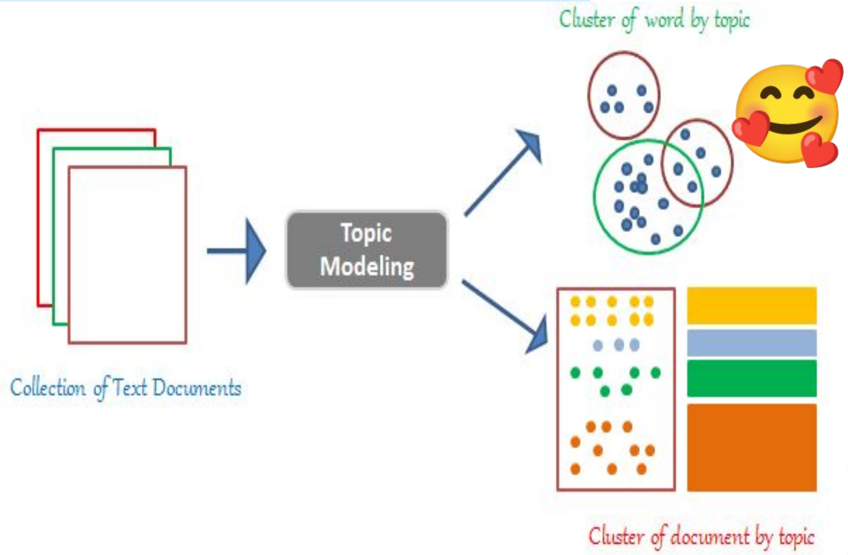
Unsupervised



[Topic Modelling With LDA -A Hands-on Introduction - Analytics Vidhya]()

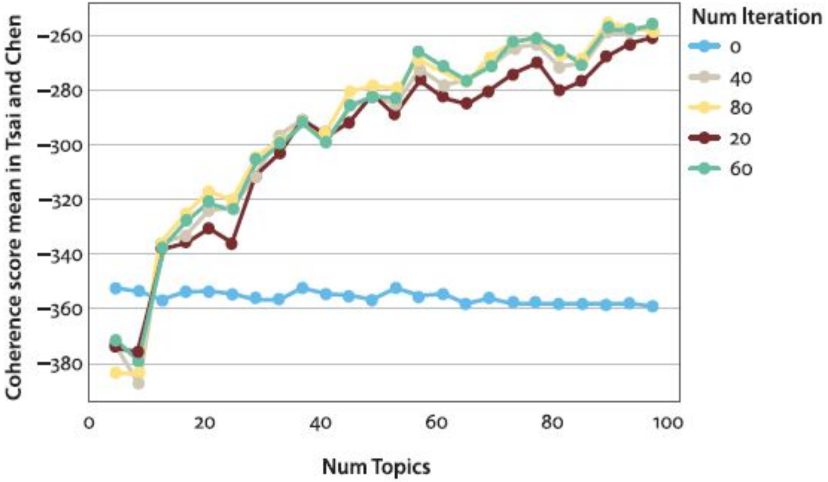
Classification tasks





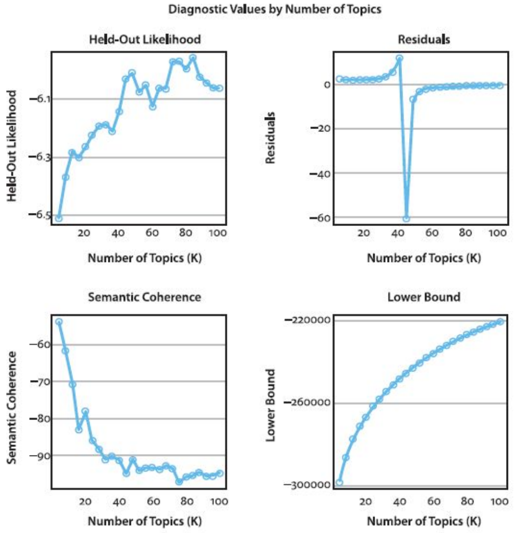
Cluster Number

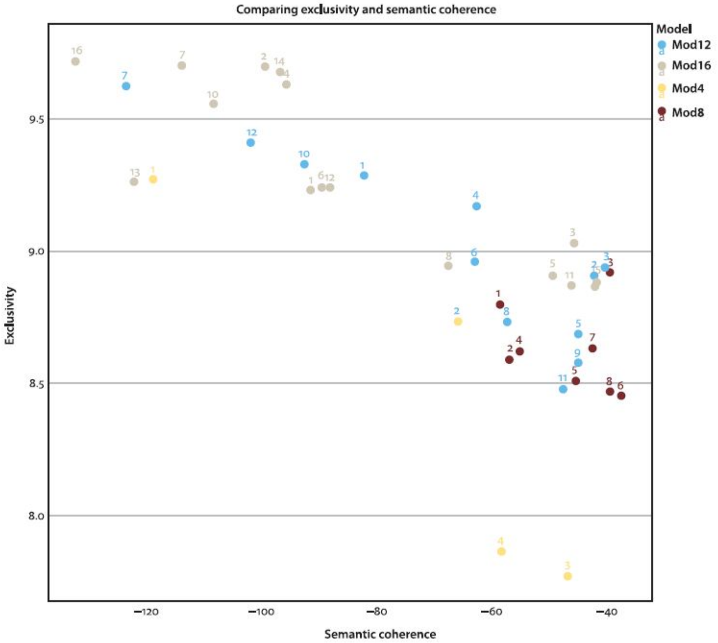
1. Perplexity
2. Semantic Coherence Metric



Cluster Number

1. **Perplexity**
2. **Semantic Coherence Metric**





Topic Modeling

Document Representation

Vector Space Model:

- Term frequency (in a doc)
- Term Frequency–Inverse Document Frequency (TF-IDF)

Document Length

Long:

- Latent Semantic Analysis (LSA): doc-topic & topic-term matrix
- Latent Dirichlet Allocation (LDA):
  Gibbs sampling:
  the distance between words in the same latent topic is minimized & the distance between words from different latent topics is maximized.

Topic Modeling

Document Length

Challenges

- Data sparsity
- Context🙈
- Labeled data🙈
- Noises 🦥🦦🦭🐋

Short:

- Sentence-LDA:
  Each document is inferred from only one topic
- Global Word Co-Occurrences Based Methods:
  Documents with similar context tend to share the same topics
- FastText-based Sentence-LDA:
  FastText associates each word with a group of similar words with a similarity degree or weight.

The main hypothesis of our proposed model is that a document can be about several topics.

Case Study

[Using Topic Modeling and Word Embedding for Topic Extraction in Twitter - ScienceDirect]

LDA+word2vec: 🧛 🧛 □ 🧟‍♀️ **K-means:**

Topic-doc matrix **A label of doc**

Challenges

- Data sparsity 🧛 🧛
- Context (ambiguity)
- Labeled data🙈
- Noises 🧛 🧛

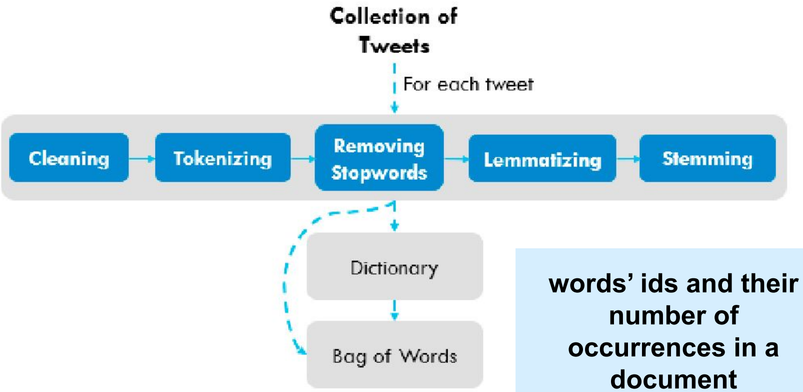


Fig. 2. Document preprocessing

words' ids and their number of occurrences in a document

Case Study

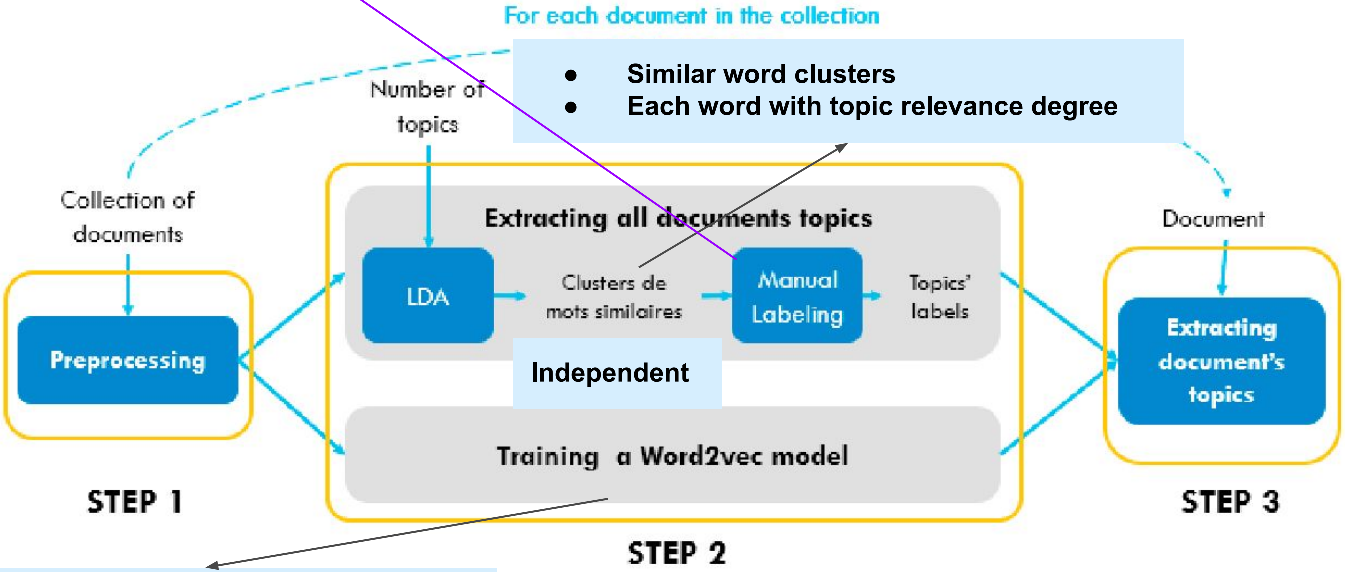
Challenges

- Data sparsity
- Context (ambiguity)
- Labeled data
- Noises

- **Label the topic of word clusters**
- **Each topic with most representative words**

- **Similar word clusters**
- **Each word with topic relevance degree**

Independent



- **Each word: a vector of similar words with similarity scores**

Fig. 1. Model architecture

Case Study

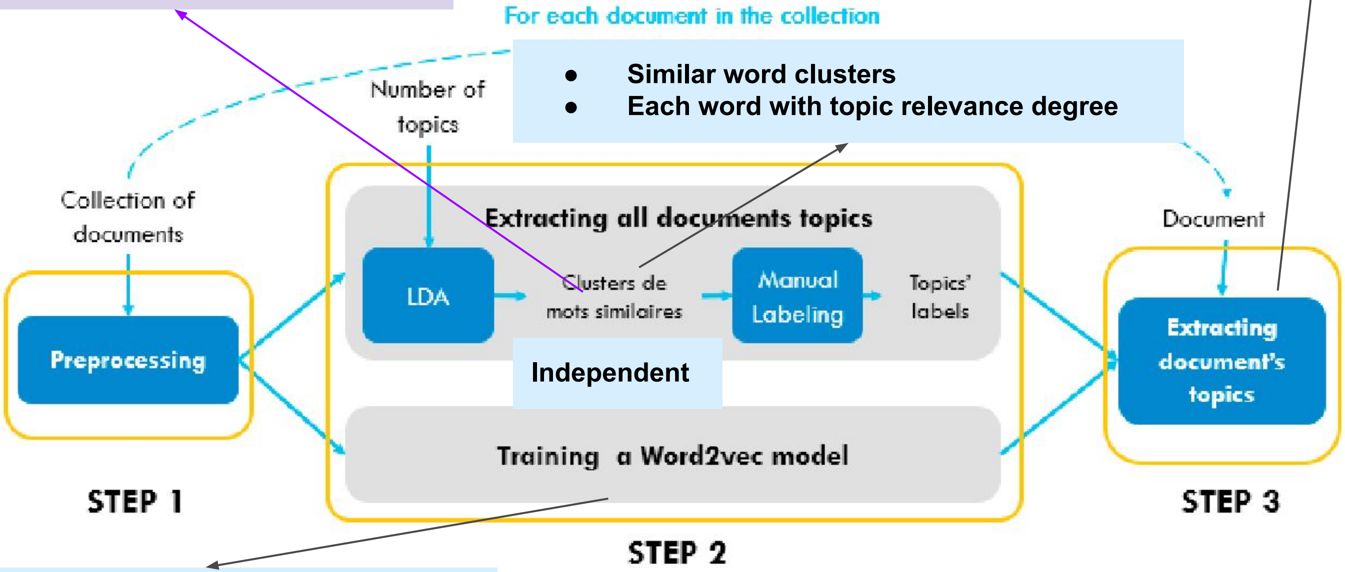
- Label the topic of word clusters
- Each topic with most representative words

Measure a document doc's score in a topic t : their similarity

$$word2vec.Similarity(representative\_words(d), representative\_words(t))$$

- Similar word clusters
- Each word with topic relevance degree

Independent



- Each word: a vector of similar words with similarity scores

Fig. 1. Model architecture

Case Study

[Using Topic Modeling and Word Embedding for Topic Extraction in Twitter - ScienceDirect]

LDA+W2V: 🦹‍♂️ 🦹‍♂️ □ 🧟‍♀️ K-means:

Topic-doc matrix A label of doc

For the data we used 114826 tweets as our short text documents and to collect them we used Twitter API [1]. To label the topics of the collected data we used a semi-automatic annotation technique. The main idea behind the latter is to try to cluster documents or tweets, using k-means, in a way documents within the same cluster are similar to each other. After clustering our documents, we will try to take a few documents (sample) from each cluster and annotate them manually. After the manual annotation, all documents within the same cluster from which we took the sample will take the label of its correspondent sample since all documents in one cluster share similar properties (see figure3).

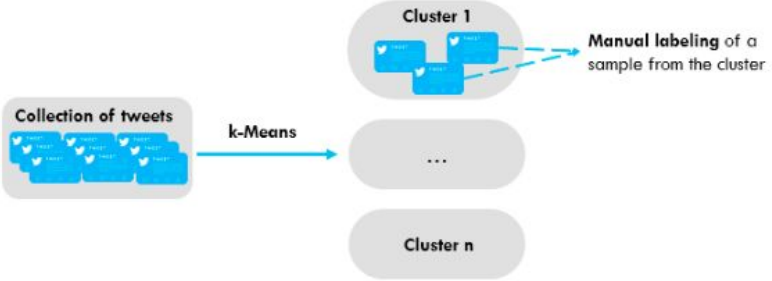


Fig. 3. Semi-automatic annotation

Case Study

[Using Topic Modeling and Word Embedding for Topic Extraction in Twitter - ScienceDirect]

LDA+W2V: 🦹 🦹 ⃞ 🧟‍♀️ K-means:

Topic-doc matrix A label of doc

After calculating the document's score in the range of the detected topics (topics detected in section 3.4.1) we obtain as a result a vector where each row represents a topic, and the column-row values represent the score of the document in the row's topic. From a topics' vector of document *d*, we will extract the most representative topics by fixing a threshold *th*. So, if the topic's score is greater than *th* so the latter is one of document *d*'s relevant topics.

$$relevant\_topics(d, th) = \{t \in Topics : withdoc\_topic\_score(d, t) \geq th\} \quad (2)$$

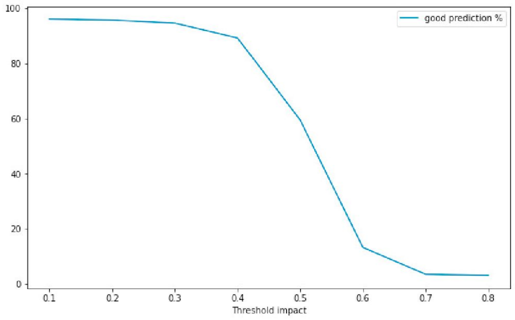


Fig. 5. Threshold impact on the percentage of good predictions

A good threshold value :

- **maximizes the percentage of good predictions**
- **minimizes the number of predictions labels per tweet**

Case Study

[Using Topic Modeling and Word Embedding for Topic Extraction in Twitter - ScienceDirect]

LDA+W2V: 🦹‍♂️ 🦹‍♂️ 🧟‍♀️ K-means:

Topic-doc matrix A label of doc

After calculating the document's score in the range of the detected topics (topics detected in section 3.4.1) we obtain as a result a vector where each row represents a topic, and the column-row values represent the score of the document in the row's topic. From a topics' vector of document *d*, we will extract the most representative topics by fixing a threshold *th*. So, if the topic's score is greater than *th* so the latter is one of document *d*'s relevant topics.

$$relevant\_topics(d, th) = \{t \in Topics : withdoc\_topic\_score(d, t) \geq th\} \qquad (2)$$

The best value to choose is 0.5

- **minimizes the number of labels "3"**
- **maximizes the threshold values (topics are relevant)**

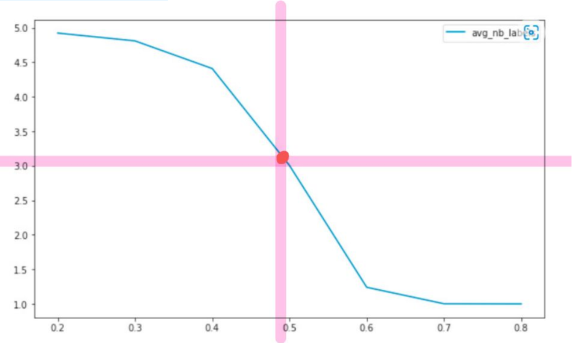


Fig. 6. Average number of labels per tweet

K means Clustering

Topic Classification

Topic Modeling