Classifying and labeling the relationships between cities with high levels of co-occurrence on the English Wikipedia

August 13, 2022

Thesis by *Diederik van Rijen* (d.w.j.vanrijen@students.uu.nl)
MSc in *Applied Data Science* at *Utrecht University*

Supervisor: *Evert Meijers*
Second supervisor: *Tongjing Wang*
Second Examiner: *Carolina Castaldi*

Universiteit Utrecht

Abstract

This study delves into three different approaches of document classification in order to successfully classify the type of relationship between European cities: "LDA Topic modeling, Word embedding classification and Word frequency representation". The first method provides a distribution of topics, the second provides hard classification, while the last method uses word frequency metrics to represent a document by its most relevant words. LDA topic modeling and word embedding classification provided very similar results for a dataset of 311.000 paragraphs, indicating a serious level of accuracy, and proving that they could both be used for classification. The paragraphs showed a pretty similar distribution of the following six topics: "entertainment, diplomacy, education, art, transportation and sport." The last method, representation of text by word frequency leaves the classification up to the viewer and can be considered visibly pleasing. The most important deliverables of this study consist of two major datasets (1). 506,328 Individually classified paragraphs that contain co-occurrences of city pairs, and 2). the aggregated classification of 1,770 city pairs belonging to 60 cities).

Keywords— European cities, network, toponym co-occurrence, classification, topic modeling, word embedding.

Contents

1 Introduction

1.1 Motivation

When we look at European cities as a network, there is a high emphasis on the nodes (cities), while the edges (relationships) are often only covered by hard numbers (based on flows of goods, transportation, connecting flights, trade data, or a wide variety of other, often rather inaccurate proxies). This can be explained by the fact that measuring relationships between cities has been a continuous and common challenge, and thus the lack of good data has been called "the dirty little secret of world city research" [27].

Luckily for us, the pursuit of good data has become easier over the past years due to the rise of data science and the steadily increase of the overall accessible data volume [16]. This offers new possibilities, combined with the fact that cities that are strongly related are often mentioned together (*'co-occur'*) in written texts, according to Meijers and Peris [23]. Analysing these toponym co-occurrences can shed new light on how these cities relate to each other. The frequency of toponym co-occurrences tells us how strongly related they are, and is relatively straightforward to obtain. The major challenge can be found in accurately classifying those relationships, in what context are cities mentioned together? Both supervised and unsupervised machine learning are plausible options for text classification, but a lexicon-based approach also seems particularly suited.

This paper is the result of a ten week long project with as overarching principal goal the exploration and mapping of the relationships between different cities and classification of these relationships. The deliverables of this project consist of this paper, a clean and reproducible code base, as well as two datasets with proper classification of both, the 1770 city pairs and the 506,328 paragraphs with city pair occurrences. This should make it possible for other researchers to validate the results and pursue this line of research. A visualisation of the classified city pairs will be presented on a European map, as well as a Lo-Fi prototype for the implementation of an interactive web application that lets the user interact with the toponym co-occurrences and relationship labels.

This project also led to the following two papers: *"The Effect of Space-Language Bias on Toponym Co-occurence Derived Networks"* by Brecht Nijman and *"Analysis of Toponym Co-occurrences on Social Media"* by Kevin O'Driscoll.

1.2 Research Question

The classification of the relationships between cities adds value to the resulting network by providing (the user with) a possible explanation on the strength of a relationship between two cities. For example, the number of co-occurrences of *Paris* and *Milan* might be due to their common link to art, and the relationship between Madrid and Manchester could potentially be attributed due to the presence of well-known football clubs. Being able to label the context of toponym co-occurrences plays a significant role in understanding their relationship. This need for accurate text classification led to the following research question:

"To what extent is it possible to classify and label the relationships between cities with a high level of linguistic co-occurrences on the English Wikipedia?"

This research question will be answered using the following sub-questions:

- What are the available options for classifying large texts?
- How useful are the available classification options for the labelling of the relationships between cities? What are the pros and cons of the different approaches?
- What is the best approach to classify and label the relationships between cities?

The remainder of this paper is organised as follows. Chapter 2 provides some background information covering some of the available classification methods, text representation and Latent Dirichlet Allocation (LDA), an unsupervised 'soft' clustering algorithm. Chapter 3 covers the data that has been used for this study and how it was preprocessed. Chapter

4 gives an insight into the methods that were used and provides reasoning for their usage. Chapter 5 presents the results and model comparison. Chapter 6, the discussion, shows the limitations of the study and provides recommendations for future work. The final chapter concludes the study and provides an answer to the research question. Supplementary information and graphical representations are to be found in the Appendix.

2 Background

2.1 Text Classification

Figure 1 is an expanded version of a figure from a literature review of 91 papers, written between 2010 an 2017, on "text classification techniques in AI" by Thangaraj et al. [32]. The visualised tree represents algorithms that were mentioned along text classification. The algorithms are divided based on their learning procedure. The methods and algorithms that are shown in the green boxes were used during this study.

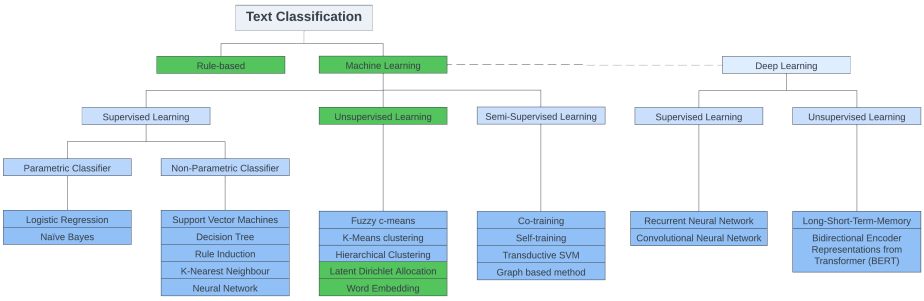


Figure 1: *Hierarchical representation of text classification algorithms.*

2.1.1 Rule-based systems

Rule-based classification systems apply (*if/then*) rules derived from elements or text patterns to determine the category of text. An example would be to set up a list of relevant words for each specific class after which the number of related words to each class gets counted. A text will receive the class with the highest count of related words. Rule based models can have good accuracy and all the steps in the classification process are clear. It, however, can take a lot of time to set up and requires deep analysis and numerous testing.

2.1.2 Machine learning-based systems

Machine learning-based classification methods have gotten more and more attention recently due to the increased computational power of computers [16]. Instead of relying on manually crafted rules, it learns to make classifications based on training data. The process of classification model training can be seen in figure 2. Pre-labeled training data is used to learn the different associations between texts in relation to their label. The next step, after acquiring the training data, is *'Feature Engineering'*. This means that features get extracted from raw data, e.g. text to numerical representation as most machine learning algorithms only understand numerical features. Features can be transformed, and a selection of desired features, that contribute to the predicted class outcome, get passed onto the machine learning model for training, which results in the final classification model. Now we can actually classify unlabeled texts by doing feature extraction, feeding those to the model and receiving a classification prediction, as shown in figure 3.

Both machine-learning and rule-based techniques can be combined in hybrid classification algorithms. The following paper by Kamruzzaman et al. demonstrates its potential, through the optimisation of a text classifier by using word relation rules instead of words, to derive a feature set from the labeled training data. Naïve Bayes Classifier is then used on those features to predict the class [21].

Figure 2: *Training a machine-learning based classification model.*

Figure 3: *Classifying text with the machine-learning based classification model.*

2.1.3 Classification Types

Classification tasks can be divided into different categories dependent on the possible classes and prediction outcome, being: *Binary, multi-class, multi-label.* The easiest type is binary classification, with only two exclusive available class labels. Popular algorithms are: '*Logistic Regression, K-Nearest Neighbours, Decision Trees, Support Vector Machine* and *Naïve Bayes*' [22]. Logistic Regression and Support Vector Machines are specifically designed for binary classification.

Multi-class classification covers the tasks that have more than two available class labels, with a single predicted class. *K-Nearest Neighbors, Decision Trees, Naïve Bayes, Random Forest, Gradient Boosting* and *Artificial Neural Networks* can all lead to good results [8].

Multi-label classification is the process where each text can be assigned multiple labels. None of the previously mentioned methods can be used directly, but some adapted versions are available: '*Multi-label Decision Trees, Multi-label Random Forests* and *Multi-label Gradient Boosting*'. Multi-label classification can also be hierarchical in nature when the labels are hierarchically structured [13], e.g. *movie (romance, comedy)* and *sport (soccer, tennis).*

2.2 Text representation and encoding

Proper feature extraction may involve encoding meaningful text into a vector representation, while preserving the context and relationships between words, in order to allow computers to decode, understand and find text patterns. Text representation can be classified into two categories: *Discrete text representations*, and *distributed or continuous text representations.*

In discrete text representation the representation of a word remains unaffected by other words and context. Known examples are one-hot encoding, basic bag-of-words (BOW), and advanced BOW (TF-IDF). Distributed text representation is when the representation of a word is dependent or not mutually exclusive of another word. Known examples are co-occurrence matrices, Word2Vec and GloVe. [12]

2.3 Latent Dirichlet Allocation

LDA, which stands for Latent Dirichlet Allocation, is an generative probabilistic model and unsupervised 'soft' clustering algorithm that uses Dirichlet distributions to find topics in a data set. LDA produces a probability distribution of groupings per item, whereas a 'hard' clustering algorithm like k-means assigns each item to a single cluster. The output from a topic model that uses LDA consist of two parts: "A distribution of words over topics, and a distribution of topics over documents." [10]. In layman's terms this means that we have words that have a higher probability of occurring for a given topic, and we have topics that have a higher probability of occurring for a given document.

To be able to find meaningful non-overlapping topics it is important to make sure that the distribution of words over topics is coherent.

3 Data

3.1 European cities

A list of 150 European cities was provided which came from the following report [15]. Due to the computational challenges of text classification the number of cities was reduced to the biggest 60 cities, based on the contiguous built-up area (morphological urban area) specified here [4], which resulted in a total of 1770 city pairs.

Table 1: *The 60 European cities with the most residents in their morphological urban area (MUA) of which the city pairs are classified.*

City	Country	Residents
Paris	France	9,591,000
London	England	8,256,000
Madrid	Spain	4,955,000
Berlin	Germany	3,776,000
Milan	Italy	3,698,000
Barcelona	Spain	3,659,000
Athens	Greece	3,331,000
Rome	Italy	2,532,000
Birmingham	England	2,363,000
Lisbon	Portugal	2,315,000
Naples	Italy	2,308,000
Katowice	Poland	2,279,000
Manchester	England	2,207,000
Hamburg	Germany	2,123,000
Budapest	Hungary	2,123,000
Bucharest	Romania	2,064,000
Warsaw	Poland	2,004,000
Stuttgart	Germany	1,735,000
Vienna	Austria	1,674,000
Munich	Germany	1,647,000
Brussels	Belgium	1,498,000
Stockholm	Sweden	1,479,000
Frankfurt	Germany	1,462,000
Cologne	Germany	1,398,000
Copenhagen	Denmark	1,360,000
Valencia	Spain	1,318,000
Turin	Italy	1,309,000
Glasgow	Scotland	1,228,000
Prague	Czech Republic	1,175,000
Lyon	France	1,175,000
Sofia	Bulgaria	1,174,000
Liverpool	England	1,170,000
Porto	Portugal	1,163,000
Seville	Spain	1,082,000
Dublin	Ireland	1,070,000
Helsinki	Finland	1,065,000
Amsterdam	The Netherlands	1,052,000
Rotterdam	The Netherlands	1,025,000
Düsseldorf	Germany	1,016,000
Essen-Oberhausen	Germany	986,000
Lille	France	953,000
Lodz	Poland	919,000
Marseille	France	862,000
Antwerp	Belgium	830,000
Bilbao	Spain	822,000
Newcastle	England	814,000
Krakow	Poland	807,000
Bochum-Herne	Germany	804,000
Thessaloniki	Greece	777,000
Nuremberg	Germany	769,000
Riga	Latvia	764,000
Duisburg	Germany	758,000
Dortmund	Germany	750,000
Hanover	Germany	747,000
ZÃ¼rich	Switzerland	718,000
Oslo	Norway	712,000
Bremen	Germany	709,000
Dresden	Germany	697,000
Sheffield	England	693,000
Palermo	Italy	680,000

3.2 Wikipedia Dump

The principal and main data source consists of all 7 million articles on the English Wikipedia (compressed and downsized to a 20GB file). However, with some adjustments and language support from *Natural Language Processing* (NLP) algorithms like SpaCy [20] (lemmatisation) and GloVe [24] (word embeddings), it should be quite easy to either include Wikipedia articles written in different languages or totally new datasets like Reddit for the classification of toponym co-occurrences.

The articles are obtained by downloading a compressed Wikipedia Dump that serves as Wikipedia back-up. The most recent Wikipedia dump for each language can be found at `https://dumps.wikimedia.org/backup-index.html`. For this study we solely used the articles from the Wikipedia dump from '2022-04-20', these articles are stored across a multitude of (.bz2) compressed Wikipedia files. The English articles can be obtained at: `https://dumps.wikimedia.org/enwiki/20220420/`.

3.2.1 Ethical considerations

According to the license information of Wikimedia all the textual content is freely available under the GNU free Documentation License and the Creative Commons Attribution-Share-Alike 3.0 License [5]. Even if some articles were to contain sensitive data this should pose no problems due to the aggregated nature of the resulting classification (classification over city pairs instead of paragraphs).

3.3 Cleaning

Because an official wikidump was used, the Wikipedia articles came in the same, straightforward format. The text contained HTML tags and external and internal links which had to be removed. A Python tool, called WikiExtractor [9], proved to be very useful as it allowed for easy extraction of the plain text from the wikipedia dump, significantly downsizing its overall size. Referral pages, pages that link to other pages and have no content themselves, were removed as they are not relevant to this study.

3.4 Preprocessing

3.4.1 Toponym co-occurrence extraction

First, the text files containing all the articles are scanned for co-occurrences for the given list of city pairs (e.g. (*Paris, London*) or (*Milan, Berlin*)), the co-occurrences of the first 380 city pairs are shown in figure 4. When a city pair has been detected, the window size of choice, in this case the full paragraph, the article title, article id and assigned paragraph id will get written as a new line to a text file corresponding to the city pair. A .csv file will also be populated for each city pair with the following columns and its values: Title, city_pair, article_id, paragraph_id and paragraph. These .csv files will play a major role in the classification process and will be turned into the resulting datasets of this study.

index	paris	london	madrid	berlin	milan	barcelona	athens	rome	birmingham	lisbon	naples	katowice	manchester	hamburg	budapest	bucharest	warsaw	stuttgart	vienna	munich
paris	0	21051	3277	7612	3251	2378	1244	6782	522	1065	1174	58	972	1327	1178	917	1548	701	5256	3062
london	21051	0	2313	7389	2512	1749	1649	4357	7537	1084	851	44	10348	1850	973	396	1223	548	4208	2334
madrid	3277	2313	0	1034	1128	5237	323	1529	105	753	314	19	843	357	205	138	261	325	724	899
berlin	7612	7389	1034	0	929	892	561	2126	280	419	304	66	415	4292	1092	404	1439	1582	5499	5097
milan	3251	2512	1128	929	0	1066	280	3818	113	246	1644	8	594	290	224	104	200	174	1179	838
barcelona	2378	1749	5237	892	1066	0	426	826	159	412	320	13	866	277	196	84	155	189	493	784
athens	1244	1649	323	561	280	426	0	1166	115	137	200	7	169	113	201	150	129	74	401	356
rome	6782	4357	1529	2126	3818	826	1166	0	153	473	3670	23	230	411	420	236	457	305	1794	1072
birmingham	522	7537	105	280	113	159	115	153	0	35	21	5	3548	82	43	12	40	62	140	127
lisbon	1065	1084	753	419	246	412	137	473	35	0	188	4	113	145	142	69	102	52	326	168
naples	1174	851	314	304	1644	320	200	3670	21	188	0	4	27	96	55	31	73	64	599	201
katowice	58	44	19	66	8	13	7	23	5	4	4	0	3	12	24	14	374	11	51	21
manchester	972	10348	843	415	594	866	169	230	3548	113	27	3	0	162	67	35	73	83	186	812
hamburg	1327	1850	357	4292	290	277	113	411	82	145	96	12	162	0	207	43	192	693	1241	1664
budapest	1178	973	205	1092	224	196	201	420	43	142	55	24	67	207	0	345	426	127	1963	406
bucharest	917	396	138	404	104	84	150	236	12	69	31	14	35	43	345	0	205	57	426	115
warsaw	1548	1223	261	1439	200	155	129	457	40	102	73	374	73	192	426	205	0	74	1041	315
stuttgart	701	548	325	1582	174	189	74	305	62	52	64	11	83	693	127	57	74	0	632	1374
vienna	5256	4208	724	5499	1179	493	401	1794	140	326	599	51	186	1241	1963	426	1041	632	0	2715
munich	3062	2334	899	5097	838	784	356	1072	127	168	201	21	812	1664	406	115	315	1374	2715	0

Figure 4: *Co-occurrence matrix of the first 20 cities from the list.*

3.4.2 Tokenisation and Lemmatisation

Second, the csv file that contains the paragraphs of a specific city pair is opened as a list of paragraphs and downsized into chunks to avoid any memory allocation errors. With the aid of priorly mentioned advanced NLP library, SpaCy, the paragraphs will get split into a sequence of tokens that each represent a word, punctuation, whitespaces, etc. The tokenizer starts by splitting the text on whitespaces, then the tokenizer processes the text from left to right. Each substring receives two checks: 1). Does the substring match a tokenizer exception rule?, and 2). Can a prefix, suffix or infix be split off?. If there is a match, the rule is applied and the tokenizer continues its loop, starting with the newly split substrings. As shown in figure 5, this allows SpaCy to split complex, nested tokens like combinations of abbreviations and multiple punctuation marks.

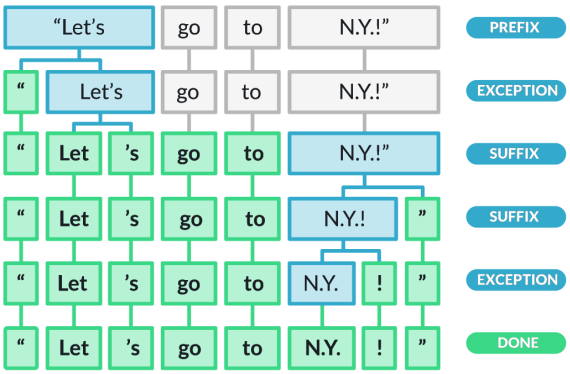


Figure 5: *Spacy tokenizer in action.* [19]

Each token gets assigned a predicted part-of-speech (POS) tag (Noun, Verb, etc.) based on the context and statistical methods. A list of possible SpaCy POS tags can be found in figure 23. The Named Entity Recognition and Dependency Parser parts were not necessarily required, and thus left out to significantly speed up the process. SpaCy has three trained pipelines that differ in the accuracy of their predictions [18]. The largest, the 'en_web_core_lg' pipeline, will be used to maximize accuracy. The available steps of this processing pipeline can be seen in Figure 6.

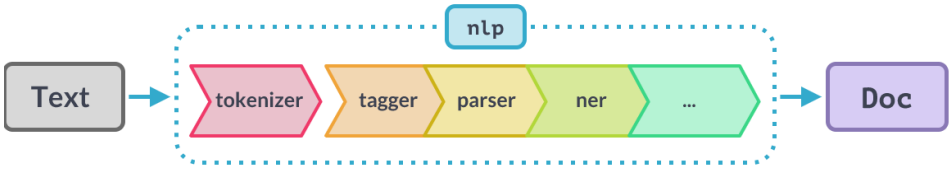


Figure 6: *spaCy processing pipeline.* [17]

Next, only the tokens within the processed text that either had a *'Noun'*, *'Verb'* or *'Adjective'* POS tag and were not considered stopwords or punctuation were included. Six new columns were created in the .csv file of each city pair, representing the clean and raw version of each of the the three different POS tags. The raw column hosting the output from the SpaCy lemmatisation and the clean column with removal of non-existent words. These lemmatisation columns, created according to the bag-of-words (BOW) model principle, disregard grammar and order but instead represent text as a histogram of word occurrences, these are extremely important as they are used for each of the three proposed classification methods. Thus logically, after finding out that the spaCy pipeline was only '97%' accurate, a list of 379.000 common English words was used to validate each word against and add in the cleaned column with only the lemmatised English words from a paragraph.

Lemmatisation was chosen over stemming due to the fact that, while lemmatisation takes a lot more time, it results in real dictionary words by taking into account context and

POS tags (e.g. *'change, changing' to change*), whereas stemmer algorithm's hard-coded rules to chop off suffixes result in stemmed versions of a word (e.g. *'change, changing' to chang*).

3.4.3 Word Frequency Metrics

Because the paragraphs were converted into a bag of lemmatised words the right frequency metric should be used in order to make the classification as accurate as possible. The easiest method to represent a document is by a simple bag-of-words representation, where a set of vectors contains the count of word occurrences. These counts are only meaningful when taking into account the length of the document, which is why the relative frequency of words will be calculated by dividing it by the number of words within a document (TF). Next, it is crucial to find how important a word is to a certain document relative to the collection of documents, with as primary goal to leave out meaningless words that occur often like, for example, 'city, year and time'. The numerical statistic that achieves this, Term Frequency-Inverse Document Frequency (TF-IDF, consists of two parts: Looking at the relative term frequency of a word in a document, and the number of documents that contain the word. Words that appear in a lot of the documents will thus be seen as less important to a specific document and normalised to a lower value [25].

4 Research Methodology

Due to the nature of this classification task, three different methods that could potentially be successful at finding a good representation over the city pairs were chosen. LDA Topic Modeling, Word Embedding Classification, and Word Frequency Metrics. All three use a very different technique and represent the relationship between two cities differently.

4.1 Translation to a data science problem

To be able to say something about the feasibility of the classification and labeling of the relationships between cities with a high level of (linguistic) co-occurrences on the English Wikipedia we have to approach it as a data science problem. What algorithms exist, or can be created to classify the relationships between cities, and do these consistently provide us with accurate results? This will be done by comparing the results from both the topic modeling and word embedding classification method between themselves as well as by manually reviewing samples of classified paragraphs. In order to keep this study relevant to the field of data science the classification labels will not be based on literature, instead topics will be sought after by an unsupervised technique that is used to find (hidden) groupings in data, called Latent Dirichlet Allocation.

4.2 LDA Topic Modeling

The discussed topics within the text belonging to each city pair were exposed through a method called LDA topic modeling. Topic modeling is an unsupervised machine learning technique that is capable of scanning a set of provided documents, detect word and phrase patterns within them and automatically cluster word groups that best characterize these set of documents and create topic distributions for these documents. The input and outputs of a topic model are visualised in Figure 7. According to Tarifa et al. [31], there are two main reasons to perform topic modeling, these are: "Selecting meaningful words to represent each topic, and having separate topics by maximising the cluster inter-distance resulting in the most distinct topics". There are a lot of different topic modeling approaches (like Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (pLSA), Latent Dirichlet Allocation (LDA) and deep learning-based lda2vec), but they all rely on the same basic assumptions: "Each **document** consists of a mixture of specified number of **topics**, and each **topic** consists of a collection of **words**". Figure 7 shows the topic modeling pipeline. The topic modeling model that was used in this study, LdaMallet [26] differs from other LDA topic models by using Gibbs sampling instead of variational (Bayes) inference. This gives better results over time, but takes longer to run [28]. Furthermore variational Bayes inference is irredeemably biased whereas the bias of Gibbs sampling approaches 0 as long as enough samples are taken.

LdaMallet uses training documents for a proper estimation of word-topic and document-topic distributions for both training documents and new documents. If topics are distinct enough it should be possible to label them and thus finding out if and what topics are dominant for each document.

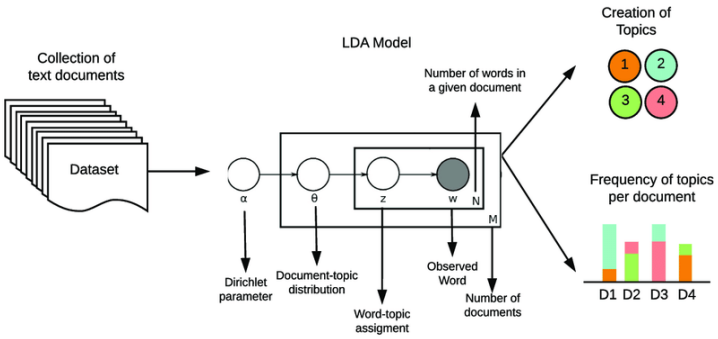


Figure 7: *Basic principle of (LDA) topic modeling.* [14]

4.2.1 Document choice

Deciding what a document represents in an lda topic model is crucial. Should paragraphs that belong to specific toponym co-occurrences be counted as a single document, or should each paragraph be counted as an individual document? It might make sense to use combine the paragraphs belonging to the co-occurrences of a city pair into a single document, in order to classify the city pair relationship straight away with the LDA topic model. However, due to the artificial creation of these city pair text files and lack of semantic connectivity between each paragraph, an LDA topic modeling algorithm might not be able to find semantically meaningful topics in them. The paragraphs containing the co-occurrences of each city pair were already extracted and merged together as a .txt file, hence, an attempt was made to find meaningful topics across the city pairs of top five biggest cities with the paragraphs of each city pair as a document. The result can be seen in figure 12. The LDA model was unable to represent cohesive topics, which can be explained due to the lack of coherency within each document. Each document consisted of a large number of paragraphs from different articles that cover a lot of different topics instead of having truly dominant topics.

After a fruitless attempt the second option was tested, where each document represents a single paragraph that will receive its own topic distribution. the count of dominant topics in the paragraphs of a city pair should then result in a classification of the city pair itself. This gives a lot more flexibility by making it possible to use classification thresholds and discarding smaller paragraphs in favour of the quality of the final clusters, Figure 13 shows how the above choice of document led to a successful LDA topic model.

4.2.2 Parameter settings

LDA topic modeling requires a given number of clusters (=topics) and will then try to find good word-topic and document-topic distributions. Because no specific number of topics is expected in these documents, an LDA model was trained for each number of topics within the range 2 to 21. Two other important parameters of an LDA model, MIN_DF and MAX_DF received different inputs to optimise the final model. The final model will leave out words that either appear in less than 5% of the documents or more than 90% of the documents. The number of iterations was set to 1000 and the optimisation interval was kept at 10 as advised by the documentation [6]. Both the Alpha and Beta hyperparameters that represent document-topic-density and topic-word density were left unchanged as these will get optimised automatically every N iterations where N is the optimisation interval. A low Alpha forces documents to only contain 1 or a few topics and a high Alpha means that documents are likely to contain a mixture of many topics, Figure 8 shows the likely distribution of topics for a document based on different Alpha values. With a high Beta the topics are composed of a large number of words from the corpus and with a low Beta the topics have less words.

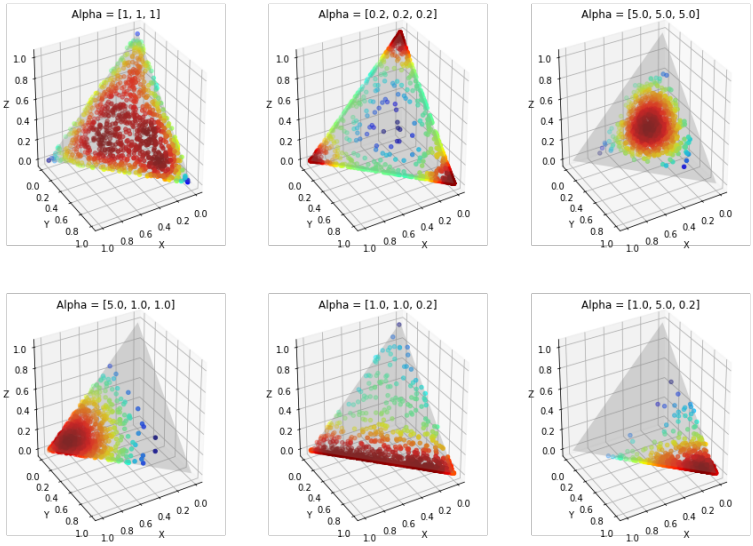


Figure 8: *Impact of the different Alpha values on the topic distribution within a document.* [2]

4.2.3 Model Performance Analysis

Each LDA model was analysed with the aid of two great tools for finding out how good a topic model is, being the topic model coherence score and visualisation of the clustered topics.

The coherence score of a topic model helps distinguish between topics that are semantically interpretable topics and topics that are artifacts of statistical inference. It is obtained by calculating the average of the distances between words for each topic to measure the degree of semantic similarity between high scoring words of a topic. Well-known NLP library *'Gensim'* provides us with a coherence model, with great parameter flexibility, that returns a coherence score. There are multiple coherence measures that can be used (e.g. C_{UMass}, C_V, C_{UCI}, C_{NPMI}), of which C_V was selected, considered the best performing coherence measure according to the following paper about the space of topic coherence measures [29]. A high coherence score indicates coherent topics, a good score to aim for would be 0.6-0.7.

Visualisation was done through *pyLDAvis* [30], a library for interactive topic model visualisation, which uses *Principal Component Analysis* (PCA) to reduce N-dimensional vectors to 2D vectors to map the clusters in a two dimensional field. A good topic model will have relatively big, similarly sized and non-overlapping bubbles scattered throughout the chart. Greater distances between the clusters represents a larger semantic difference, similarly sized bubbles are a sign that the topics are equally represented, and large circles mean that the topics are well represented in the documents. By paying attention to these three characteristics we could get an accurate representation of the dominant topics of our documents and decide whether these clusters represent good, meaningful topics. By experimenting with the relevance metric slider for each topic coherent, interpretable list of words can be created that can easily be labeled based on the common topic. Table 3 shows the six clusters with their 15 most relevant words and their assigned labels that were found by the LDA model in the paragraphs of the 435 city pairs.

After approving the topic model both visually and through its coherence score the topic distribution of the documents should be analysed (e.g. table 4), to avoid unwanted distributions. Some topics might be more present than others by having words with high term frequencies that aren't necessarily representative of a meaningful topic. If multiple topics have a similar score for a paragraph, this indicates that there is no true dominant topic and thus should not result in a dominant topic. By looking at the distribution of the values for each dominant topic we can chose a classification threshold that is required in order for a topic to be deemed dominant and the classification to be trustworthy. Hard classification will be done, which means that majority Voting on the classified paragraphs will be done based on equal weights.

4.3 Word Embedding Classification

4.3.1 Adressed Problem

A major drawback of LDA topic modeling is the lack of control a user has on the topics, due to the unsupervised learning nature of the technique. To leave out a topic, one could lower the number of topics by one and hope for good results, this however just create a new set of clusters and topic distributions. Figure 13 shows quite a bit of overlap between topic 3 (Art) and 4 (Education) which could lead to inaccurate categorisation due to the potential similarity of the topics. The clustering might be considered good, but could be better if either Art or Education was left out or merged together. LDA topic modeling makes this difficult because the distribution of a topic depends on all other topics. The Word Embedding classification method does not have this problem, as it requires a list of provided topics and simultaneously allows the user to leave out (and add) topics they want. Figures 15a and 15b show this in practice, the topic modeling and word embedding algorithms had significantly more similar results for *'Art'* classified documents when its closest neighbouring topic *'Education'* was left out.

4.3.2 Word Embedding

Word embedding, a term used for distributed representations of text in an n-dimensional space, shows words as a multi-dimensional (e.g. 300) vector that tries to encode the semantics of a word such that words closer in the vector space are expected to be similar in meaning, also seen as the similarity principle. Figure 9 shows the basic principle of word embeddings and their strength as they are essential for solving most NLP problems. For example, it allows a user to quickly find the words that are close to one another according to the word embedding model, or do mathematical calculations with words (e.g. *'king'* - *'man'* = *'queen'*). The word embeddings also work as feature extraction method as it transforms raw data (characters) into a (meaningful) numerical representation that is required by most machine learning algorithms. Existing word embeddings have been trained on a large number of texts, where each word is represented by a point in the embedding space. These points are learned and moved around based on the words that surround the target word in the texts that the word embedding model is trained on. While the representation of a word by a n-dimensional vector might not totally cover the semantics of that word, we rely on the principle that it will result in an accurate overall classification of a certain document if enough words are used to find overall similarity. One thing to note would be that documents with a low number of words, or city pairs with a lower number of documents are at risk of having 'wrong' classification predictions. Figure 16a and 16b show how paragraphs with a lower number of lemmatised words are less likely to lead to the same classification across all topics in both the LDA topic model and word embedding algorithm.

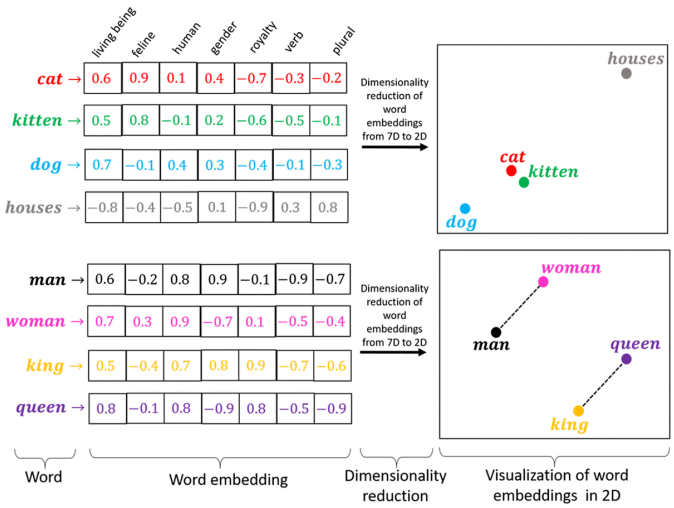


Figure 9: *Basic principle of word embeddings.* [3]

4.3.3 Topic Vector Creation

If the position of the defined topics in the vector space could be found, the words within a paragraph could be categorised by looking which topic vector is closest to them, and thus classify the paragraph based on the sum of the distances between the words and their closest topic. Due to the fact that words are 'only' represented along 300 dimensions, a larger spatial distance between the vectors of the different topics highly increases the accuracy of the predictions of this model. The hard part is to find the vector that covers the semantic meaning of a certain topic. This 'perfect' representation can be pursued by taking the mean of the vector representation of multiple self-selected topic keywords, however, finding the right keywords turned out to be very hard to do manually. It ended up being a lot better, for the use case of labeling city pairs, to use the 15 most relevant words from each cluster that was found through the unsupervised LDA modeling visualised by pyLDAvis with a preferably lower relevancy value. This works because the relevancy of these words to their topic has already been proven and words that often appear together are likely to belong to similar topics. Because both the LDA topic model and word embedding classification model use the same topics their results can be compared to each other. The performance of this recommended approach is documented in the following section.

4.3.4 Pre-trained word embedding

To continue with this method a word embedding is required, either self-trained or pre-trained. The word embedding classification model is word embedding independent and thus any word embedding can be provided as parameter. However, Pre-trained word embeddings are often trained on very large datasets leading to a pretty accurate spatial vector representations of a word [24]. For this study a pre-trained word embedding by GloVe was chosen that has been trained on 840 billion uncased tokens from Common Crawl data, resulting in a vocabulary of 2.2 million words by GloVe, which consists of a 5GB text file that can be found on the project page of GloVe under the following name: 'glove.840B.300d.zip' [24].

4.3.5 Design choices

To further improve the accuracy of the model paragraphs that either have less than 10 words or were assigned a dominant topic with a score below 0.7 by the LDA topic model were not used for the classification of the city pairs. This improves the results because the model will not be able to find a dominant topic in a paragraph if there is none and because small sized paragraphs are less accurately categorised according to Figure 16b. The words in a paragraph can also be left out of the classification equation for various reasons, being: Not having a vector representation, not having a higher similarity score to the most similar topic than the similarity threshold or being ambiguous (i.e. having a score gap between the most dominant topic and the number two and three topics that is too small). Ax example of this can be seen in figure 10. The similarity threshold and allowed score gap are parameters that can be changed. The similarity scores between each word of a document and its most similar topic are temporarily saved, after which the topic with the highest overall similarity score is selected as the predicted classification of a document, see figure 11.

```
==========================================================================================

word:           'publish'
choice:         kept

==>     category:, art,
        similarity score: 0.3980032801628113

scores:
        ----category----          ----score----
        art                     0.3980032801628113
        education               0.3359798789024353
        diplomacy               0.2570980191230774
        transportation          0.2010655403137207
        entertainment           0.17235074937343597
        sport                   0.14829422533512115

==========================================================================================

word:           'catholic'
choice:         discarded
reasoning:      ambiguity

scores:
        ----category----          ----score----
        education               0.37401846051216125
        art                     0.3628559708595276
        diplomacy               0.3564857542514801
        entertainment           0.23469537496566772
        sport                   0.18883058428764343
        transportation          0.18536171317100525

==========================================================================================

word:           'zodiacal'
choice:         discarded
reasoning:      low similarity score

scores:
        ----category----          ----score----
        art                     0.11023600399494171
        education               0.04033316671848297
        entertainment           0.028182532638311386
        diplomacy               0.017856841906905174
        transportation          -0.02329024113714695
        sport                   -0.044465430080890656

==========================================================================================
```

Figure 10: *Example of word-topic similarity as internal process in the embedding classification algorithm.*

```
[('game', (0.7163149118423462, 'sport')),
 ('year', (0.5055342316627502, 'sport')),
 ('country', (0.5844901204109192, 'diplomacy')),
 ('flight', (0.6627249717712402, 'transportation')),
 ('capital', (0.47764599323272705, 'diplomacy')),
 ('year', (0.5055342316627502, 'sport')),
 ('lion', (0.3837367296218872, 'art')),
 ('season', (0.6950360536575317, 'sport')),
 ('athlete', (0.44466254115104675, 'sport')),
 ('sign', (0.42370912432670593, 'art')),
 ('play', (0.636486291885376, 'sport')),
 ('play', (0.636486291885376, 'sport')),
 ('professional', (0.46715009212493896, 'education'))]

[('art', 0.8074458539485931), ('diplomacy', 1.0621361136436462), ('education', 0.46715009212493896),
('entertainment', 0), ('sport', 4.140054553747177), ('transportation', 0.6627249717712402)]

The most dominant topic is 'Sport'
```

Figure 11: *Example of document (=paragraph) classification by our word embedding algorithm.*

4.3.6 Algorithm Performance Analysis

For a large number of documents, the predicted topic by the word embedding classification model will be compared to the predicted dominant topic of the LDA topic model. It is expected that paragraphs with a dominant topic with a distribution score of over 0.7 will

get the same predicted topic from the word embedding model. The role of the paragraph length and topic score on the accuracy of the model will be analysed, as well as the topic dependency of the accuracy of the word embedding classification model.

4.4 Word Frequency Representation

Classification can be subjective and paragraphs don't necessarily have a dominant topic. To avoid mislabeling (e.g. misinterpretation) and information reduction through topic labeling the last method will leave the interpretation of a city pair's topic(s) up to the individual by representing the context of the co-occurrences through various word frequency metrics. These words and frequencies could be displayed by any chart that can appropriately fit this type of date, i.e. wordclouds and (horizontal) bar charts.

This method relies on predefined word list with a numerical statistic of choice (absolute term frequency, relative term-frequency, term frequency-inverse document frequency). The difficulty lies in leaving out just enough 'useless' words through stopword removal and general word frequency threshold boundaries in order to keep a set of meaningful words that correctly represent the topics belonging to a city pair. If done correctly this methods representation should show a certain number of words that can be interpreted by the user and should portray the topics that belong to a specific city pair.

4.5 Differences between the Classification techniques

Table 2: *Feature differences between the three classification techniques.*

Features	**Classification Techniques**		
	LDA Topic Model	*Word Embedding Model*	*Word Frequency Representation*
Learning	Unsupervised	semi-supervised	Unsupervised?
Classification Type	Multi-label	Multi-class	None
Special Input	Number of required Topics	Topic Keywords	Collection of Documents
Representation	Distribution of Topics	Category	Wordcloud

5 Results

In this section, the results are presented, these include topic distributions by the LDA topic model, topic classification by the word embedding classification model, the word frequency representation and a comparison between the two implemented models.

5.1 LDA Topic Models

5.1.1 Distribution of words over topics

Figure 12 and 13 show the pyLDAvis visualisation of two of the major LDA topic models that were created. The first figure shows the visualisation with the merged paragraphs of the first 10 city pair as a document, which resulted in very flawed topics of uneven sizes, that were highly overlapping and had little semantic meaning. The second figure shows the final LDA topic model that was used for both the topic distributions of the 311.000 paragraphs belonging to a total of 435 city pairs and to create topic vectors for the word embedding classification model. The only negative trait of this model is that topic 4 (Education) and topic 3 (Art) seem to overlap a little, but they do hold very semantically different words as can be seen in Table 3 which would be a good reason to keep them both.

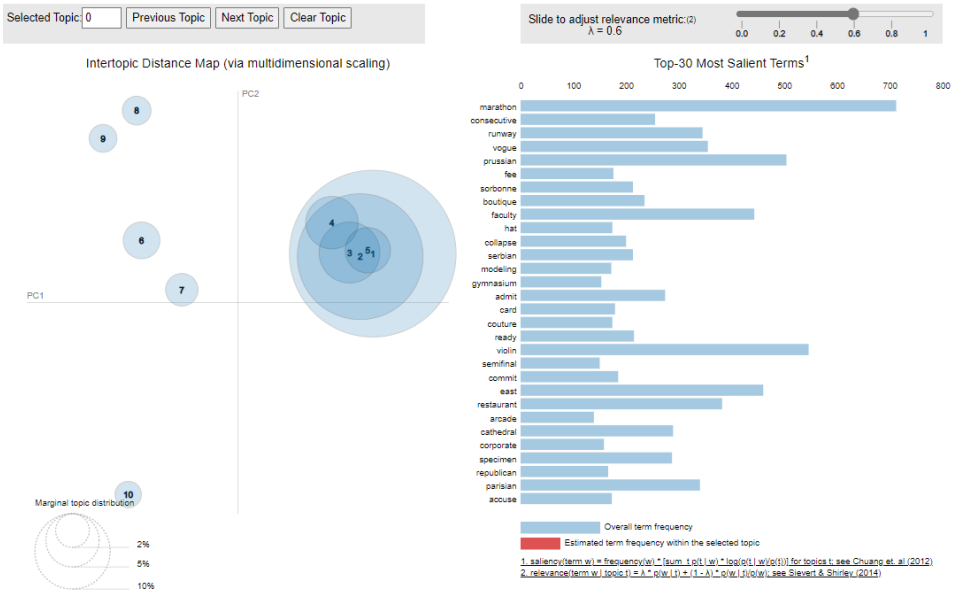


Figure 12: *LDA model pyLDAvis visualisation with the merged paragraphs of each city pair as a document. 51.000 paragraphs, 10 city pairs.*

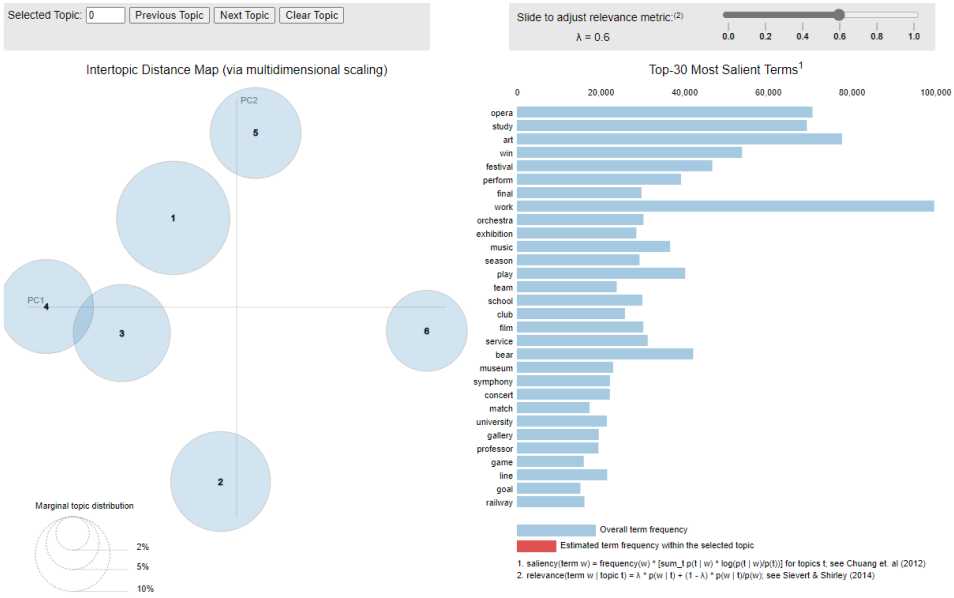


Figure 13: *LDA model pyLDAvis visualisation with the paragraphs of the city pairs as their own document. 311.000 paragraphs, 435 city pairs.*

Table 3 shows the words that are most important to each topic which were obtained by putting the relevance parameter slider on 0.2. A relevance value of 0 shows words that are truly relevant to a topic, but often have a very low frequency counts, where as a relevance value of 1 shows the words by frequency, which obviously shows high frequency words that are more likely to occur in multiple topics. By selecting 0.2, we make sure that we get the words that are mostly based on their relevance to a specific topic, but also to the entire number of documents. The number between parentheses corresponds to the topic numbers in Figure 13, and the given topic labels could be changed at any time as this has no influence on the resulting topic distribution or word embedding predictions. The mean of the vectors of these 15 most relevant words to a topic were also used to get the topic vectors for the word embedding classification model.

Table 3: *Distribution of words over topics for the 30 biggest cities and their city pairs (relevance parameter = 0.2).*

	(1) Diplomacy	(2) Entertainment	(3) Art
1	War	Opera	Exhibition
2	Embassy	Festival	Art
3	Army	Perform	Museum
4	Diplomatic	Orchestra	Gallery
5	Ambassador	Symphony	Exhibit
6	Treaty	Concert	Painting
7	Protest	Music	Collection
8	Force	Film	Paint
9	Mission	Sing	Portrait
10	Arrest	Theatre	Artist
11	Government	Performance	Sculpture
12	Police	Role	Fashion
13	Attack	Premiere	Design
14	Party	Tour	Contemporary
15	Minister	Band	Painter

	(4) Education	(5) Transportation	(6) Sport
1	Study	Railway	Final
2	School	Route	Win
3	Professor	Line	Team
4	University	Operate	Match
5	Graduate	Flight	Game
6	Bear	Station	Goal
7	Degree	Service	Club
8	Research	Airline	League
9	College	Airport	Champion
10	Teach	Train	Championship
11	Science	Passenger	Season
12	Education	Speed	Score
13	Philosophy	Aircraft	Round
14	Doctorate	Rail	Tournament
15	Faculty	Network	Football

5.1.2 Distribution of topics over documents

Table 4: *Distribution of dominant topics by LDA topic modeling in the documents (=paragraphs).*

Topic	**Paragraphs**	
	Raw Count	*Percentage (%)*
All	310,828	100%
Entertainment	65,366	21.03%
Diplomacy	57,828	18.60%
Education	55,784	17.95%
Art	52,721	16.96%
Transportation	47,008	15.12%
Sport	32,121	10.33%

Table 4 shows the raw and relative number of paragraphs that were labeled with each of the six topics by that were found by the LDA topic model. The most common topic among the 310.828 paragraphs is 'entertainment' with 21,03% of the documents, closely followed by the topics 'diplomacy', 'education', 'art' and 'transportation'. 'Sport' is the least common topic with 10,33% of the paragraphs. This even distribution shows that the topics occur equally often. Having a topic that is highly present like in the LDA model of figure 12 might be an indication of a general topic that would not result in interesting labels if you only look at the most dominant topic of a document. E.g. Labeling 95% of your dataset of news articles as news is probably not the specificity you were looking for when you also have topics about politics, economics, conflicts, environmental issues. The key take away here is to either select the right amount of specificity for your clusters (topics) or allow multi-labeling.

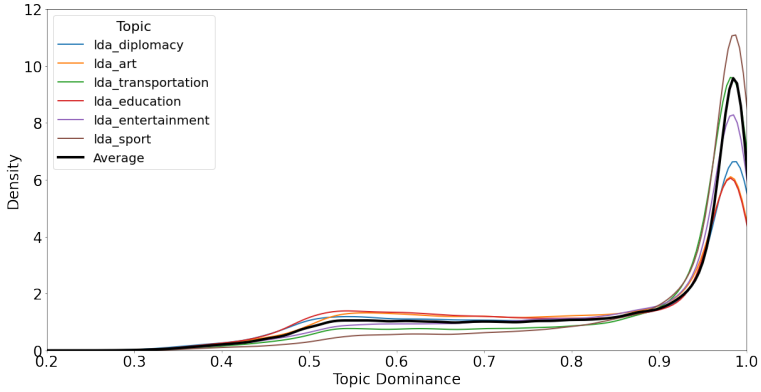


Figure 14: *Density of the scores of dominant topics, grouped by topic.*

Most documents have very high scoring dominant topics according to Figure 14, meaning there is only one very dominant topic, and thus multi-labeling might be unnecessary and could potentially even be counterproductive in classification tasks.

5.1.3 Distribution of topics over city pairs

Table 5: *Disitribution of dominant topics by LDA topic modeling for 1628 of the 1770 city pairs.*

Topic	**City Pairs**	
	Raw Count	*Percentage (%)*
All	1,628	100%
Entertainment	474	29.15%
Sport	432	26.57%
Transportation	302	18.57%
Education	154	9.47%
Diplomacy	141	8.67%
Art	123	7.56%

Table 6: *Distribution of topics across three of the city pairs predicted by the LDA topic model.*

barcelona_manchester			**paris_milan**			**warsaw_prague**		
Topic	*Count*	*Percentage*	*Topic*	*Count*	*Percentage*	*Topic*	*Count*	*Percentage*
all	694	100	all	2222	100	all	601	100
sport	612	88.18	art	758	34.11	diplomacy	274	45.59
transportation	29	4.18	entertainment	497	22.37	transportation	97	16.14
entertainment	19	2.74	transportation	295	13.28	entertainment	81	13.48
diplomacy	10	1.44	diplomacy	218	9.81	education	65	10.82
education	12	1.73	education	178	8.01	art	51	8.49
art	12	1.73	sport	276	12.42	sport	33	5.49

The LDA topic modeling results of 3 of the 435 city pairs can be seen in Table 6. Due to the relatively even representation of topics across most city pairs we show the city pair classification as a distribution. This means that each city pair will have a column that displays the most dominant topic as well as counts for each individual topic. This makes multi-labeling of city pairs possible, and facilitates certain tasks, like seeing all city pairs that have the topic Art above a certain percentage threshold, without having to be the most dominant topic.

5.1.4 Analysis

Although it may time and good parameter tuning before a good result is obtained with a topic model, the process is straight forward and an example of soft classification, it gives a distribution of topics per document, which gives the user the flexibility to either keep the soft classification or use their own threshold for hard classification.

The most time consuming part when performing LDA topic modeling is finding the right number of topics for the unsupervised clustering algorithm to find good results. If the wrong number of topics is selected the model will force words into clusters that semantically might not make sense, this is clearly visible in the topic model visualisation displayed in Figure 12. The user either needs to try out multiple topic counts combined with a multitude of different settings or requires prior information on the number of clusters that should be found. It is also impossible to leave out a specific cluster without changing the word-topic distribution or document-topic distributions, because the distribution of topics is dependent on all topics.

The content of the clusters are also highly dependent on the documents that are given as input. If the documents contain a lot of fashion related words, a fashion related topic will likely be formed as it presents a dominant cluster, whereas the model might not cluster an important smaller sized topic due to the presence of bigger potentially less meaningful clusters. This was clearly visible when comparing the LDA topic models of the 5 and 30 biggest cities, the former had a way more defined fashion cluster which could be explained because the Paris and Milan were present in a relatively higher number of city pairs.

On top of that LDAMallet does not know how to deal with word sense disambiguity (WSD) which can negatively affect the topics. For example, the word bank could have different meanings in different documents (i.e. bank as organisation or bank of a river). The following paper by Jordan Boyd-Graber et al. addresses this problem and introduces a topic model that includes WSD as a hidden variable , unfortunately no working model was provided [11].

5.2 Word Embedding Classification Model

5.2.1 Document (paragraph) classification

Table 7: *Distribution of the dominant topics by word embedding classification in the documents (=paragraphs).*

Topic	**Paragraphs**	
	Raw Count	*Percentage (%)*
All	310,828	100%
Diplomacy	68,981	22.19%
Entertainment	64,498	20.75%
Education	51,413	16.54%
Art	44,715	14.39%
Transportation	42,557	13.69%
Sport	38,664	12.44%

Table 7 shows the raw and relative number of paragraphs that were labeled with each of the six topics by the word embedding algorithm. The most common topic among the 310,828 paragraphs is *'entertainment'* with 22.09% of the documents, the remainder of the topics are pretty evenly distributed between 12 and 21% with *'Sport'* as smallest with 12.44%. The only difference in topic placement between the two models is that *'Diplomacy'* came above *'Entertainment'* in the word embedding classification. Unexpectedly high scoring topics could be attributed to 'general' topic vectors which are closer to a lot of words with low semantic value in a document or due to a similarity threshold parameter that is set too low. The former option could result in documents with a lack of expressive topics that would be more likely to be classified to the general topic. Leaving 'meaningless' words out of the classification of a paragraph would increase the accuracy of the model, but would be hard to achieve due to the subjective and ambiguous semantic value of a word. One negative result of the word embedding classification model would be that it will always assign a topic, not classifying when the similarity threshold has not been matched is possible but requires more research as the similarity between a topic and a word can differ a lot.

5.2.2 City pair classification

Table 8: Distribution of dominant topics by the word embedding classification algorithm for 1628 of the 1770 city pairs.

Topic	**City Pairs**	
	Raw Count	*Percentage (%)*
All	1,628	100%
Sport	457	28.11%
Entertainment	404	24.85%
Art	317	19.50%
Transportation	296	18.20%
Diplomacy	108	6.64%
Education	44	2.71%

Table 9: *Distribution of topics across three of the city pairs predicted by the word embedding classification algorithm.*

barcelona_manchester			**paris_milan**			**warsaw_prague**		
Topic	*Count*	*Percentage*	*Topic*	*Count*	*Percentage*	*Topic*	*Count*	*Percentage*
all	694	100	all	2222	100	all	601	100
sport	608	87.60	art	637	28.67	diplomacy	291	48.42
transportation	29	4.12	entertainment	492	22.14	transportation	85	14.14
entertainment	16	2.31	sport	330	14.85	entertainment	80	13.31
diplomacy	17	2.45	diplomacy	286	12.87	education	65	10.82
education	14	2.02	transportation	261	11.75	art	40	6.66
art	10	1.44	education	216	9.72	sport	40	6.66

The word embedding classification algorithm results of 3 of the 435 city pairs can be seen in Table 9.

5.2.3 Analysis

The 300 dimensions of the word representations and sheer number of vectors in the word embedding model make it very hard to correctly display the topics in a two-dimensional space, although it is possible through Principal Component Analysis. This visual representation could help determine whether these topic vectors may allow clear classification. Topics that are further apart from each other are more likely to correctly classify documents.

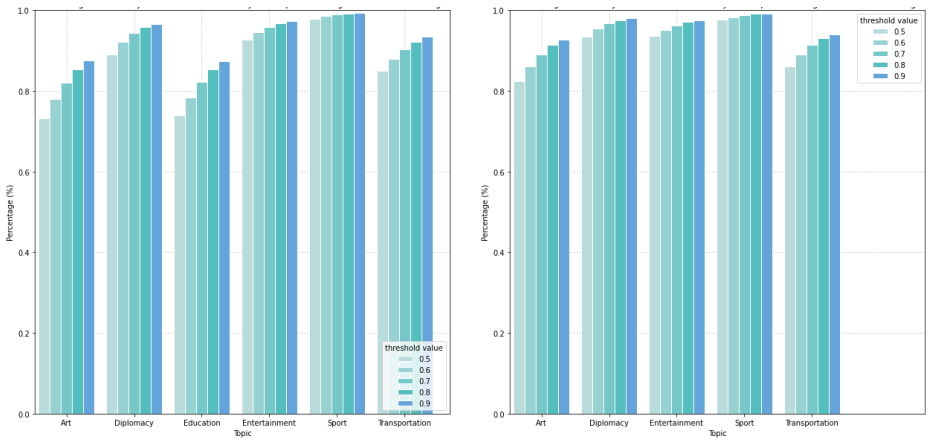
Words that either are semantically meaningless or are word sense ambiguous (bear=could be a verb and animal, degrees=could be angle, temperature metric, educational level) are still represented in a word embedding model and might negatively impact the classification of a document as long as their similarity score to a topic is high enough. It is very hard to define a clear minimal similarity and ambiguity threshold that negates these words while leaving impactful words unaffected. Right now the algorithm always leads to a classification, even when the scores might not be as high.

5.3 Model Comparison

Table 10: *Bin distribution of LDA topic modeling scores for differently classified documents (paragraphs).*

	Documents		
Score Bin	*All*	*Different classification (raw)*	*Different classification (%)*
[0-1]	310,828	57,600	18.53
[0.9-1.0]	162,233	13,049	8.04
[0.8-0.9]	36,067	5,202	14.42
[0.7-0.8]	33,458	7,231	21.61
[0.6-0.7]	31,646	9,850	31.13
[0.5-0.6]	29,996	12,775	42.59
[0.4-0.5]	14,427	7,662	53.11
[0.3-0.4]	2,909	1,767	60.74
[0.2-0.3]	92	68	69.57
[0.1-0.2]	0	0	0.00
[0.0-0.1]	0	0	0.00

Table 10 shows the topic distribution score belonging to the most dominant topic of the paragraphs that were differently classified by the word embedding classification model. The more prevalent the dominant topic of a paragraph, according to the LDA topic model, the more likely it is to receive the same classification by the word embedding classification algorithm. By using the right threshold for classification we can lower the 'error' rate and still take into account most of the paragraphs. Differences start to get bigger below 0.6, which is totally logical because lower values indicate the presence of multiple topics that can result in multiple valid classification options. A bottom threshold of 0.7, for example, keeps the paragraphs with very dominant topics, would result in 231,758 (74.6% of the total) paragraphs that are available for city pair classification and would significantly lower the percentage of different classified paragraphs from 18.53% to 11.0%.

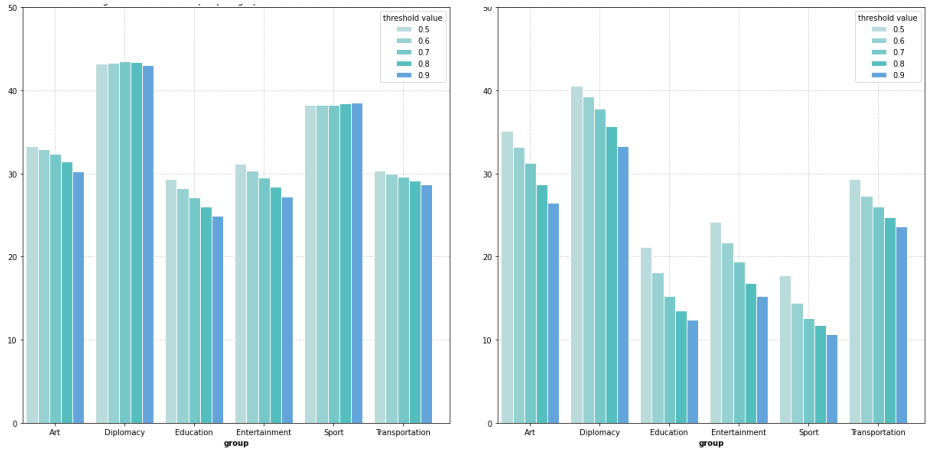


(a) *With education.* **(b)** *With education left out.*

Figure 15: *The percentage of paragraphs that received the same classification in both techniques, grouped by topic and threshold value.*

Figure 15 shows how similar the classification results are between the LDA topic model and word embedding classification model, grouped by topic and set of different thresholds. The classification similarity of certain topics like *'Sport'* do not seem to be as affected by different threshold values when compared to other topics, however a higher topic distribution score threshold will always result in increased similarity between the prediction by both models. Figure 15b shows how this similarity can be increased by leaving out specific topics. In this case *'education'*, one of the two topics that were relatively close to

each other according to the visualisation of LDA topic model clusters in 13, was left out, which let to a higher level of classification similarity.



(a) *For the same classification.* **(b)** *For a different classification.*

Figure 16: *Average number of lemmatised words per paragraph, grouped by topic and threshold value.*

Figure 16a and 16b show how the number of lemmatised words of a paragraph impacts the classification accuracy. All topics are less accurately classified with less words, but some topics (i.e. *'Sport'*) are more impacted than others. Discarding paragraphs with less than 10 words seemed to increase the accuracy significantly, while the model accuracy seemed to be unaffected by higher bottom threshold values.

5.4 Word Frequency Representation

5.4.1 Wordclouds

Figure 17: *TF-IDF representation of the 30 most relevant words to Paris-Milan.*

Figure 18: *TF-IDF representation of the 30 most relevant words to Barcelona-Manchester.*

Figure 19: *TF-IDF representation of the 30 most relevant words to Warsaw-Prague.*

6 Discussion

6.1 Limitations

Due to the wide range of Wikipedia article topics and the common nature between city pairs, quite often these city pairs have quite similar topics, which makes it hard to find significant differences in topic. A low number of toponym co-occurrences also makes it a lot harder to confidently label their relationship, a solution to this would be to leave out the classification for city pairs with less than a certain number of co-occurrences. Although we have shown here that it is in fact possible to label the relationship between cities its accuracy is still highly dependent on the distance between topic clusters or vectors, the further apart they are the easier it is to use them for classification. For example, paragraphs that received *'Art'* as dominant topic in the LDA topic model quite often were labeled differently by the word embedding classification algorithm, as seen in figure 15. This can be explained due to the minimal distance between the topics *'Art'* (3) and *'Education'* (4) in figure 13. The paragraphs with *'sport'* had the highest number of same classified paragraphs for both classification methods, which would be expected when we see how isolated the cluster (6) is in figure 13.

One major limitation of both methods is that they don't work well with documents that should not be assigned a class. This is due to the fact that an LDA topic model provides a distribution of topics and word embedding classification model provides hard classification by finding out the closest topic vector to the words of a document. Future work should look at setting up a good threshold method to provide both models with the ability to give accurate 'no class' predictions.

Due to the computational power required for the preprocessing and classification a relatively small number of cities and their city pairs have been classified. This limitation could be countered by using multiprocessing, which unfortunately was impossible on our Windows device [7].

6.2 Future Work

This study takes a step in the right direction by looking at multiple ways to classify and label the paragraphs where the city pairs are mentioned. This study serves as a proof of concept, by successfully labeling 1770 city pairs of the 60 biggest European cities. The dataset should definitely be extended further by running the code more efficiently, accompagnied with more processing power and more time.

Further analysis could yield better results by experimenting with the window size, right now it uses the full paragraph of the city pair co-occurrences, but this might not be the most effective window size. The results might also get more accurate by putting different threshold values in the word embedding classification model to the test. The lemmatisation of the documents could also be improved by including the Name Entity Resolution (NER) and Dependency Parser components. This would make it possible to use weights in the word embedding classification algorithm based on a word's NER label (e.g. Google = organisation) or grammatical position in a sentence (e.g. assign subjects higher weights). Making it possible to leave meaningless words out of the classification process.

7 Conclusion

The goal of this research was to explore and evaluate the possible methods of classification and labeling of the relationships between cities with a high number of co-occurrences on the English Wikipedia, and to end up with a qualitative dataset with proper classification of a large number of city pairs. Three methods were selected, explored and tested, each with their own set of pros and cons. A total of 60 cities were used as input, resulting in 1770 classified city pairs. The remainder of the city pairs (90 extra cities and 9.405 city pairs) could be classified in a similar manner, only requiring more time and computational power. Some city pairs had very clear topics as shown in figure 6, some city pairs had no dominant topics, while the classification of other city pairs was hindered by the small number co-occurrences. All in all, this study could be seen as a success as it shows that classification on this scale is totally possible and a viable option for this use case.

LDA topic modeling is good for labeling documents and finding initial topics and can be used to set up the topic vectors that are required for the word embedding classification model to perform hard classification. The results in this paper and the associated datasets are proof that it is possible to classify the relationships between cities with a high level of linguistic co-occurrences on the Englis Wikipedia.

This study provides as a stepping stone for further progress in the field of the classification of relationships of city pairs. However, the scientific relevancy of this approach is not only limited to city pairs, as you could look for the co-occurrences of any word pair (e.g. Hitler-Churchill, slavery-USA, Corona-lungs, etc.), and on top of that classify their relationship. The word embedding classification algorithm has the potential to be just as accurate as the LDA topic model that it was build on, as long as the topics are well defined and far enough from each other. However, the beauty of this method is that it makes use of unsupervised clustering to find topics and a very controlled method, word embedding classification, to label the different documents (paragraphs in this case).

References

[1] Topic modeling and mallet installation guide. https://programminghistorian.org/en/lessons/topic-modeling-and-mallet, visited on 08-07-2022.

[2] Part 2: Topic modeling and latent dirichlet allocation (lda) using gensim and sklearn. https://www.analyticsvidhya.com/blog/2021/06/part-2-topic-modeling-and-latent-dirichlet-allocation-lda-using-gensim-and-sklearn/, visited on 09-07-2022.

[3] Word embedding: Basics. https://medium.com/@hari4om/word-embedding-d816f643140, visited on 09-07-2022.

[4] Morphological objects: Mua (morphologic urban areas). https://database.espon.eu/db2/jsf/DicoSpatialUnits/DicoSpatialUnits_html/ch02s02.html, visited on 10-07-2022.

[5] Wikimedia license information. https://dumps.wikimedia.org/legal.html, visited on 17-05-2022.

[6] mallet_lda: A wrapper function for lda using the mallet machine learning toolkit. https://rdrr.io/github/matthewjdenny/SpeedReader/man/mallet_lda.html, visited on 23-06-2022.

[7] Multiprocessing - process-based parallelism. https://docs.python.org/3/library/multiprocessing.html, visited on 27-04-2022.

[8] M. Aly. Survey on multiclass classification methods. *Neural Netw*, 19(1):9, 2005.

[9] G. Attardi. Wikiextractor. https://github.com/attardi/wikiextractor, 2015.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[11] J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, pages 1024–1033, 2007.

[12] R. Cartuyvels, G. Spinks, and M.-F. Moens. Discrete and continuous representations and processing in deep learning: looking forward. *AI Open*, 2:143–159, 2021.

[13] R. Cerri, R. C. Barros, and A. C. De Carvalho. Hierarchical multi-label classification using local neural networks. *Journal of Computer and System Sciences*, 80(1):39–56, 2014.

[14] X. Chen, D. Zou, G. Cheng, and H. Xie. Detecting latent topics and trends in educational technologies over four decades using structural topic modeling: A retrospective of all volumes of computers education. *Computers Education*, 151:103855, 2020.

[15] E. M. Committee et al. Espon project 1.4. 3. *Study on Urban Functions. Final Report. Luxemburg, Luxemburg: Office for Official Publications of the European Communities*, 2007.

[16] K. A. I. Hammad, M. A. I. Fakharaldien, J. M. Zain, and M. Majid. Big data analysis and storage. In *International Conference on Operations Excellence and Service Engineering*, pages 10–11, 2015.

[17] M. Honnibal. Spacy processing pipeline., year = visited on 04-07-2022, howpublished = https://spacy.io/usage/processing-pipelines.

[18] M. Honnibal. Available trained pipelines for english. https://spacy.io/models/en, 2022.

[19] M. Honnibal. How tokenisation works. https://spacy.io/usage/linguistic-features#how-tokenizer-works, 2022.

[20] M. Honnibal and I. Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[21] S. Kamruzzaman and F. Haider. A hybrid learning algorithm for text classification. 09 2010.

[22] R. Kumari and S. K. Srivastava. Machine learning: A review on binary classification. *International Journal of Computer Applications*, 160(7), 2017.

[23] E. Meijers and A. Peris. Using toponym co-occurrences to measure relationships between places: review, application and evaluation. *International Journal of Urban Sciences*, 23(2):246–268, 2019.

[24] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[25] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. New Jersey, USA, 2003.

[26] R. Řehůřek and P. Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[27] J. Rennie Short, Y.-H. Kim, M. Kuus, and H. WELLS. The dirty little secret of world cities research: Data problems in comparative analysis. *International Journal of Urban and Regional Research*, 20:697 – 717, 10 2009.

[28] L. G. D. A. Rivera, A. Ilin, and T. Raiko. Comparison of variational bayes and gibbs sampling in reconstruction of missing values with probabilistic principal component analysis. *Pahikkala, Väyrynen, Kortela and Airola (eds.)*, page 25, 2010.

[29] M. Röder, A. Both, and A. Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408, 2015.

[30] C. Sievert and K. Shirley. Ldavis: A method for visualizing and interpreting topics. 06 2014.

[31] A. Tarifa, A. Hedhili, and W. L. Chaari. A filtering process to enhance topic detection and labelling. *Procedia Computer Science*, 176:695–705, 2020.

[32] M. Thangaraj and M. Sivakami. Text classification techniques: A literature review. *Interdisciplinary Journal of Information, Knowledge, and Management*, 13:117, 2018.

Appendices

Appendix A - Code

A.1 Packages

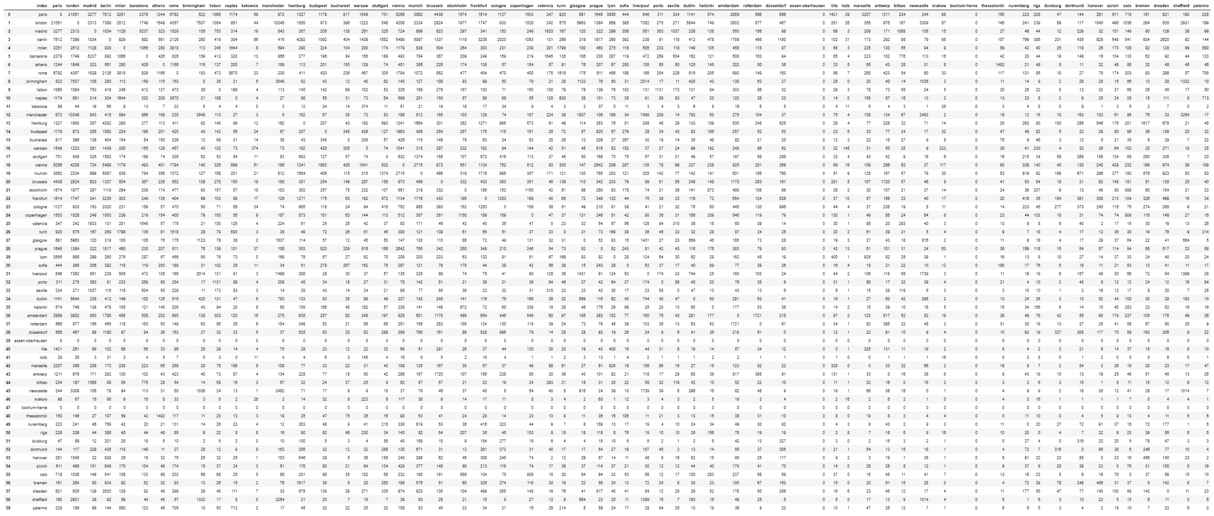
Python version 3.9.13 was used for this research. All used packages and required versions can be found in table 11.

Table 11: *Python Packages.*

Package	Version	Additional Info
en_core_web_lg	3.3.0	(SpaCy pipeline)
gensim	**3.8.3(!)**	Installation Guide [1]
ipywidgets	7.7.0	
nltk	3.7	
notebook	6.4.12	
numpy	1.22.4	
pandas	1.4.2	
Pillow	9.1.1	
pip	22.1.2	
seaborn	0.11.2	
sklearn	0.0	
spacy	3.3.1	
tqdm	4.64.0	
wordcloud	1.8.1	
wordfreq	3.0.1	

A.2 Datasets

A.2.1 Co-occurrence Matrix



	index	paris	london	madrid	berlin	milan	barcelona	athens	rome	birmingham	lisbon	naples	katowice	manchester	hamburg	budapest	bucharest	warsaw	stuttgart	vienna	munich	brussels	stockholm	frankfurt	cologne	copenhagen	valencia	turin	glasgow	prague	lyon	sofia	liverpool	porto	seville	dublin	helsinki	amsterdam	rotterdam	düsseldorf	essen-oberhausen	lille	lodz	marseille	antwerp	bilbao	newcastle	krakow	bochum-herne	thessaloniki	nuremberg	riga	duisburg	dortmund	hanover	zürich	oslo	bremen	dresden	sheffield	palermo
0	paris	0	21051	3277	7612	3251	2378	1244	6782	522	1065	1174	58	972	1327	1178	917	1548	701	5256	3062	4438	1574	1614	1127	1503	347	920	681	1645	3555	444	646	311	334	1141	574	2959	566	556	0	1421	29	2207	1211	234	244	98	0	150	223	228	47	144	281	611	718	161	821	180	228
1	london	21051	0	2313	7389	2512	1749	1649	4357	7537	1084	851	44	10348	1850	973	396	1223	548	4208	2334	2824	1977	1747	933	1528	242	575	5963	1384	868	385	7362	275	271	5844	749	3802	877	467	0	261	26	356	976	187	3309	67	0	146	241	239	59	117	1045	496	1036	264	635	2931	189
2	madrid	3277	2313	0	1034	1128	5237	323	1529	105	753	314	19	843	357	205	138	261	325	724	899	623	297	341	153	249	1833	167	120	322	299	306	551	363	1037	238	138	550	156	98	0	69	3	209	171	1068	105	15	0	27	46	44	12	229	32	101	149	90	129	38	99
3	berlin	7612	7389	1034	0	929	892	561	2126	280	419	304	66	415	4292	1092	404	1439	1582	5499	5097	1337	1119	3235	2020	1053	131	269	319	1817	260	392	238	91	118	412	478	1798	456	1180	0	102	31	173	292	98	79	95	0	107	769	386	201	435	928	648	541	934	2620	92	144
4	milan	3251	2512	1128	929	0	1066	280	3818	113	246	1644	8	594	290	224	104	200	174	1179	838	504	264	303	231	239	201	1798	100	460	275	118	505	233	118	149	105	458	118	97	0	68	3	235	130	55	94	9	0	59	42	40	25	118	26	170	108	92	128	59	580
5	barcelona	2378	1749	5237	892	1066	0	426	826	159	412	320	13	866	277	196	84	155	189	493	784	367	209	249	159	219	1645	135	106	230	287	119	472	269	504	162	121	508	163	94	0	65	4	223	102	776	113	16	0	42	20	44	18	148	16	104	133	62	92	44	123
6	athens	1244	1649	323	561	280	426	0	1166	115	137	200	7	169	113	201	150	129	74	401	356	226	174	135	57	154	57	81	76	207	67	250	105	85	50	125	145	202	50	36	0	33	5	65	43	28	31	8	0	1492	21	49	5	11	32	49	96	32	48	45	46
7	rome	6782	4357	1529	2126	3818	826	1166	0	153	473	3670	23	230	411	420	236	457	305	1794	1072	852	477	404	470	400	175	1518	175	611	456	188	186	254	229	515	205	690	149	163	0	96	7	268	423	54	60	33	0	117	131	85	10	27	75	174	203	93	298	57	705
8	birmingham	522	7537	105	280	113	159	115	153	0	35	21	5	3548	82	43	12	40	62	140	127	108	93	99	50	79	21	28	1123	76	50	31	2014	17	11	420	43	138	63	27	0	25	0	20	40	14	1036	3	0	11	14	9	2	28	25	15	66	13	28	1332	10
9	lisbon	1065	1084	753	419	246	412	137	473	35	0	188	4	113	145	142	69	102	52	326	168	278	157	103	71	150	100	79	78	139	76	102	131	1131	173	131	94	303	86	32	0	29	3	78	73	58	24	5	0	20	26	22	5	12	32	37	68	26	45	17	50
10	naples	1174	851	314	304	1644	320	200	3670	21	188	0	4	27	96	55	31	73	64	599	201	150	57	89	68	55	125	630	38	101	73	35	91	89	83	47	20	120	26	23	0	14	0	158	87	16	13	2	0	13	23	8	2	9	25	24	25	15	111	8	713
11	katowice	58	44	19	66	8	13	7	23	5	4	4	0	3	12	24	14	374	11	51	21	19	18	17	34	9	4	3	2	37	0	11	3	4	3	6	8	16	6	5	0	4	11	5	4	3	1	26	0	3	4	8	0	8	1	3	5	2	7	0	2
12	manchester	972	10348	843	415	594	866	169	230	3548	113	27	3	0	162	67	35	73	83	186	812	185	103	129	74	187	224	38	1837	106	169	34	7499	209	14	793	60	275	104	37	0	75	4	108	134	67	2482	2	0	19	12	16	10	163	103	61	86	76	33	3264	17
13	hamburg	1327	1850	357	4292	290	277	113	411	82	145	96	12	162	0	207	43	192	693	1241	1684	301	352	1271	965	573	61	49	114	353	76	51	298	45	29	133	109	635	248	526	0	28	4	77	228	32	71	14	0	25	263	90	100	256	546	176	201	1617	678	21	45
14	budapest	1178	973	205	1092	224	196	201	420	43	142	55	24	67	207	0	345	426	127	1963	406	254	257	175	118	161	25	72	57	820	57	279	28	34	43	83	166	257	52	53	0	23	6	33	77	24	17	32	0	47	48	82	5	32	29	80	96	35	139	20	32
15	bucharest	917	396	138	404	104	84	150	236	12	69	31	14	35	43	345	0	205	57	426	115	149	75	63	24	63	25	26	12	209	27	267	30	18	14	35	48	82	21	20	0	12	3	22	18	27	9	8	0	75	9	46	3	12	5	21	35	5	29	7	22
16	warsaw	1548	1223	261	1439	200	155	129	457	40	102	73	374	73	192	426	205	0	74	1041	315	287	232	192	94	144	42	51	45	816	52	152	37	27	24	99	182	248	86	52	0	22	145	31	50	25	9	223	0	35	41	230	4	32	36	64	102	20	271	18	25
17	stuttgart	701	548	325	1582	174	189	74	305	62	52	64	11	83	693	127	57	74	0	632	1374	158	107	872	416	113	37	45	50	168	70	75	57	31	21	49	57	197	68	298	0	33	4	43	42	9	18	6	0	16	215	34	58	268	158	134	68	260	335	7	20
18	vienna	5256	4208	724	5499	1179	493	401	1794	140	326	599	51	186	1241	1963	426	1041	632	0	2715	973	551	1134	752	512	83	300	147	2642	209	287	135	78	98	237	238	925	201	359	0	69	16	109	286	62	37	117	0	90	338	140	40	130	240	429	232	196	974	39	109
19	munich	3062	2334	899	5097	838	784	356	1072	127	168	201	21	812	1684	406	115	315	1374	2715	0	466	319	1716	985	357	171	121	130	765	203	121	325	142	77	142	141	601	156	760	0	51	9	125	167	57	79	30	0	53	819	82	159	671	296	377	180	576	923	53	83
20	brussels	4438	2824	623	1337	504	367	226	852	108	278	150	19	185	301	254	149	287	158	973	466	0	332	433	393	351	46	139	110	342	233	79	89	51	55	248	148	1170	283	161	0	291	5	167	1720	57	46	9	0	41	53	54	18	31	62	148	181	51	135	25	40
21	stockholm	1574	1977	297	1119	264	209	174	477	93	157	57	18	103	352	257	75	232	107	551	319	332	0	195	152	1150	42	61	86	260	63	178	74	21	36	141	872	489	105	86	0	26	2	30	107	21	37	14	0	24	36	207	9	13	46	90	889	60	104	21	23
22	frankfurt	1614	1747	341	3235	303	249	135	404	99	103	89	17	129	1271	175	63	192	872	1134	1716	433	195	0	1283	189	40	65	72	349	132	44	75	38	23	116	72	654	124	528	0	37	10	57	156	23	40	17	0	20	416	35	154	381	308	213	104	326	499	15	34
23	cologne	1127	933	153	2020	231	159	57	470	50	71	68	34	74	965	118	24	94	416	752	985	393	152	1283	0	159	36	51	49	210	81	38	41	21	32	76	60	445	130	996	0	44	4	37	238	19	8	11	0	14	223	45	277	373	245	116	75	274	265	9	21
24	copenhagen	1503	1528	249	1053	239	219	154	400	79	150	55	9	187	573	161	63	144	113	512	357	351	1150	189	159	0	47	37	131	246	51	42	90	36	31	166	339	546	119	79	0	130	1	49	85	24	64	8	0	23	44	103	18	31	74	74	909	116	148	27	16
25	valencia	347	242	1833	131	201	1645	57	175	21	100	125	4	224	61	25	25	42	37	83	171	46	42	40	36	47	0	23	32	54	67	56	128	94	310	38	18	60	39	14	0	30	1	88	20	293	40	3	0	13	9	8	6	40	2	17	15	30	19	13	26
26	turin	920	575	167	269	1798	135	81	1518	28	79	630	3	38	49	72	26	51	45	300	121	139	61	65	51	37	23	0	31	73	169	38	38	46	23	32	28	97	24	25	0	20	2	61	39	21	5	4	0	9	7	18	4	17	12	36	30	19	76	9	214
27	glasgow	681	5963	120	319	100	106	76	175	1123	78	38	2	1837	114	57	12	45	50	147	130	110	86	72	49	131	32	31	0	52	83	15	1431	27	23	659	46	185	73	28	0	19	3	27	43	18	816	2	0	11	8	18	4	17	29	37	64	22	41	564	5
28	prague	1645	1384	322	1817	460	230	207	611	76	139	101	37	106	353	820	209	816	168	2642	765	342	260	349	210	246	54	73	52	0	82	243	81	42	43	116	176	383	79	93	0	43	13	61	101	31	24	63	0	36	159	118	15	54	57	114	94	56	517	23	56
29	lyon	3555	868	299	260	275	287	67	456	50	76	73	0	169	76	57	27	52	70	209	203	233	63	132	81	51	67	169	83	82	0	28	124	64	30	62	29	152	48	19	0	400	1	925	82	25	39	1	0	16	13	8	10	27	14	37	33	24	40	20	24
30	sofia	444	385	306	392	118	119	250	188	31	102	35	11	34	51	279	267	152	75	287	121	79	178	44	38	42	56	38	15	243	28	0	53	37	17	40	89	77	39	26	0	16	4	18	21	22	10	12	0	185	17	76	5	18	11	21	53	13	41	11	17
31	liverpool	646	7362	551	238	505	472	105	186	2014	131	91	3	7499	298	28	30	37	57	135	325	89	74	75	41	90	128	36	1431	81	124	53	0	174	23	744	25	160	103	24	0	44	2	106	118	58	1739	3	0	11	19	19	6	157	48	30	66	72	54	1389	28
32	porto	311	275	363	91	233	269	85	254	17	1131	89	4	209	45	34	18	27	31	78	142	51	21	38	21	36	94	46	27	42	64	37	174	0	58	40	20	76	35	9	0	31	1	86	17	32	39	4	0	21	4	10	2	45	8	12	12	24	12	16	94
33	seville	334	271	1037	118	118	504	50	229	11	173	83	3	14	29	43	14	24	21	98	77	55	36	23	32	31	310	23	23	43	30	17	23	58	0	47	13	43	13	6	0	6	1	15	29	119	6	3	0	3	10	10	1	2	16	12	17	8	20	7	25
34	dublin	1141	5844	238	412	149	162	125	515	420	131	47	6	793	133	83	35	99	49	237	142	248	141	116	76	166	38	32	659	116	62	40	744	40	47	0	60	281	63	41	0	19	1	27	60	42	296	2	0	13	24	35	3	13	52	44	100	30	28	183	10
35	helsinki	574	749	138	478	105	121	145	205	43	94	20	8	60	109	166	48	182	57	238	141	148	872	72	60	339	18	28	46	176	29	89	25	20	13	60	0	177	53	36	0	14	2	15	39	10	16	5	0	15	34	156	6	14	15	40	283	23	52	16	19
36	amsterdam	2959	3802	550	1798	458	508	202	690	138	303	120	16	275	635	257	82	248	197	925	601	1170	489	654	445	546	60	97	185	383	152	77	160	76	43	281	177	0	1721	218	0	87	2	123	817	52	82	19	0	38	49	78	42	55	95	179	237	105	178	49	36
37	rotterdam	566	877	156	456	118	163	50	149	63	86	26	6	104	248	52	21	86	68	201	156	283	105	124	130	119	39	24	73	79	48	39	103	35	13	63	53	1721	0	61	0	34	1	82	395	22	48	3	0	31	30	19	13	37	25	41	58	67	50	25	9
38	düsseldorf	556	467	98	1180	97	94	36	163	27	32	23	5	37	525	53	20	52	298	359	760	161	86	528	996	79	14	25	28	93	19	26	24	9	6	41	36	218	61	0	0	12	1	22	81	10	9	4	0	10	92	19	327	305	117	70	59	163	206	8	22
39	essen-oberhausen	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
40	lille	1421	261	69	102	68	65	33	96	25	29	14	4	75	28	23	12	22	33	69	51	291	26	37	44	130	30	20	19	43	400	16	44	31	6	19	14	87	34	12	0	0	1	325	131	11	19	2	0	8	4	2	3	21	9	14	37	15	16	6	10
41	lodz	29	26	3	31	3	4	5	7	0	3	0	11	4	4	6	3	145	4	16	9	5	2	10	4	1	1	2	3	13	1	4	2	1	1	1	2	2	1	1	0	1	0	0	1	1	1	16	0	0	3	8	0	1	2	0	0	0	6	0	1
42	marseille	2207	356	209	173	235	223	65	268	20	78	158	5	108	77	33	22	31	43	109	125	167	30	57	37	49	88	61	27	61	925	18	106	86	15	27	15	123	82	22	0	325	0	0	33	32	66	2	0	19	8	7	2	54	3	26	16	20	23	17	47
43	antwerp	1211	976	171	292	130	102	43	423	40	73	87	4	134	228	77	18	50	42	286	167	1720	107	156	238	85	20	39	43	101	82	21	118	17	29	60	39	817	395	81	0	131	1	33	0	16	38	6	0	8	43	18	15	13	19	25	45	51	45	13	26
44	bilbao	234	187	1068	98	55	776	28	54	14	58	16	3	67	32	24	27	25	9	62	57	57	21	23	19	24	293	21	18	31	25	22	58	32	119	42	10	52	22	10	0	11	1	32	16	0	18	2	0	3	2	6	3	15	1	8	11	15	12	9	12
45	newcastle	244	3309	105	79	94	113	31	60	1036	24	13	1	2482	71	17	9	9	18	37	79	46	37	40	8	64	40	5	816	24	39	10	1739	39	6	296	16	82	48	9	0	19	1	66	38	18	0	1	0	4	9	9	2	18	35	12	41	28	17	1014	7
46	krakow	98	67	15	95	9	16	8	33	3	5	2	26	2	14	32	8	223	6	117	30	9	14	17	11	8	3	4	2	63	1	12	3	4	3	2	5	19	3	4	0	2	16	2	6	2	1	0	0	4	3	16	1	1	1	1	7	0	4	4	1
47	bochum-herne	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
48	thessaloniki	150	146	27	107	59	42	1492	117	11	20	13	3	19	25	47	75	35	16	90	53	41	24	20	14	23	13	9	11	36	16	185	11	21	3	13	15	38	31	10	0	8	0	19	8	3	4	4	0	0	11	10	0	4	5	9	13	4	11	5	8
49	nuremberg	223	241	46	769	42	20	21	131	14	26	23	4	12	263	48	9	41	215	338	819	53	36	416	223	44	9	7	8	159	13	17	19	4	10	24	34	49	30	92	0	4	3	8	43	2	9	3	0	11	0	20	27	72	91	37	16	72	177	1	6
50	riga	228	239	44	386	40	44	49	85	9	22	8	8	16	90	82	46	230	34	140	82	54	207	35	45	103	8	18	18	118	8	76	19	10	10	35	156	78	19	19	0	2	8	7	18	6	9	16	0	10	20	0	4	7	22	9	23	39	55	6	5
51	duisburg	47	59	12	201	25	18	5	10	2	5	2	0	10	100	5	3	4	58	40	159	18	9	154	277	18	6	4	4	15	10	5	6	2	1	3	6	42	13	327	0	3	0	2	15	3	2	1	0	0	27	4	0	316	23	20	5	78	47	3	3
52	dortmund	144	117	229	435	118	148	11	27	28	12	9	8	163	256	32	12	32	268	130	671	31	13	381	373	31	40	17	17	54	27	18	157	45	2	13	14	55	37	305	0	21	1	54	13	15	18	1	0	4	72	7	316	0	85	38	6	246	77	10	4
53	hanover	281	1045	32	928	26	16	32	75	25	32	25	1	103	546	29	5	36	158	240	296	62	46	308	245	74	2	12	29	57	14	11	48	8	16	52	15	95	25	117	0	9	2	3	19	1	35	1	0	5	91	22	23	85	0	23	16	456	140	23	2
54	zürich	611	496	101	648	170	104	49	174	15	37	24	3	61	176	80	21	64	134	429	377	148	90	213	116	74	17	36	37	114	37	21	30	12	12	44	40	179	41	70	0	14	0	26	25	8	12	1	0	9	37	9	20	38	23	0	76	31	100	5	19
55	oslo	718	1036	149	541	108	133	96	203	66	68	25	5	86	201	96	35	102	68	232	180	181	889	104	75	909	15	30	64	94	33	53	66	12	17	100	283	237	58	59	0	37	0	16	45	11	41	7	0	13	16	23	5	6	16	76	0	37	59	15	10
56	bremen	161	264	90	934	92	62	32	93	13	26	15	2	76	1617	35	5	20	260	196	576	51	60	326	274	116	30	19	22	56	24	13	72	24	8	30	23	105	67	163	0	15	0	20	51	15	28	0	0	4	72	39	78	246	456	31	37	0	142	6	7
57	dresden	821	635	129	2620	128	92	48	298	28	45	111	7	33	678	139	29	271	335	974	923	135	104	499	265	148	19	76	41	517	40	41	54	12	20	28	52	178	50	206	0	16	6	23	45	12	17	4	0	11	177	55	47	77	140	100	59	142	0	11	23
58	sheffield	180	2931	38	92	59	44	45	57	1332	17	8	0	3264	21	20	7	18	7	39	53	25	21	15	9	27	13	9	564	23	20	11	1389	16	7	183	16	49	25	8	0	6	0	17	13	9	1014	4	0	5	1	6	3	10	23	5	15	6	11	0	6
59	palermo	228	189	99	144	580	123	46	705	10	50	713	2	17	45	32	22	25	20	109	83	40	23	34	21	16	26	214	5	56	24	17	28	94	25	10	19	36	9	22	0	10	1	47	26	12	7	1	0	6	6	5	3	4	2	19	10	7	23	6	0

Figure 20: *Co-occurrence of the 1770 city pairs corresponding to 60 different cities.*

A.2.2 LDA Classified Paragraphs

	paragraph_id	city_pair	paragraph	merged_POS	indexer	lda_diplomacy	lda_transportation	lda_education	lda_art	lda_sport	lda_entertainment	lda_dominant	lda_dominant_score	index_id
0	1	berlin_milan	after his tenure in academia, he continued to ...	[tenure, academia, month, year, travel, incide...	0	0.646392	0.001234	0.001288	0.324088	0.000847	0.026151	lda_diplomacy	0.646392	0
1	2	berlin_milan	one of the astronomers selected for the search...	[astronomer, search, priest, invitation, group...	1	0.000920	0.018644	0.000922	0.977949	0.000606	0.000958	lda_art	0.977949	1
2	3	berlin_milan	there are plenty of air connections between ye...	[plenty, air, connection, city, connection, cl...	2	0.001468	0.992814	0.001471	0.001752	0.000967	0.001529	lda_transportation	0.992814	2
3	4	berlin_milan	since 2009, 'the brandery', an urban fashion s...	[fashion, year, language, monitor, ranking, wo...	3	0.002832	0.439793	0.002837	0.549726	0.001865	0.002948	lda_art	0.549726	3
4	5	berlin_milan	when considering the commuter belts or metropo...	[commuter, belt, area, datum, population, orde...	4	0.004585	0.889086	0.004593	0.005473	0.003019	0.093243	lda_transportation	0.889086	4
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
506297	32	paris_marseille	Since 1849, the Paris-Marseille railway passes...	[railway, northeast, station, line, rail, link...	506297	0.001898	0.990708	0.001902	0.002266	0.001250	0.001976	lda_transportation	0.990708	506297
506298	72	hamburg_bremen	After the war, direct trade was minimal. What ...	[war, trade, port, city, tobacco, rice, cotton...	506298	0.299221	0.657007	0.001032	0.001230	0.040437	0.001073	lda_transportation	0.657007	506298
506299	301	london_zürich	At Vassar College Mann developed lifelong frie...	[friendship, student, role, history, psycholog...	506299	0.024290	0.001148	0.532335	0.370914	0.000788	0.070525	lda_education	0.532335	506299
506300	84	munich_nuremberg	Promoted to SS-'Oberführer', Scheel on 25 Apri...	[promote, base, transfer, encompass]	506300	0.941534	0.011555	0.012065	0.014376	0.007930	0.012540	lda_diplomacy	0.941534	506300
506301	84	stuttgart_nuremberg	Promoted to SS-'Oberführer', Scheel on 25 Apri...	[promote, base, transfer, encompass]	506301	0.012043	0.941046	0.012065	0.014376	0.007930	0.012540	lda_transportation	0.941046	506301

506302 rows × 14 columns

Figure 21: *Distribution of LDA topic modeling topics for the 506.302 paragraphs that belong to the 1770 city pairs of the 60 cities.*

A.2.3 Classified City Pairs

	city_pair	paragraphs	lemmatised_paragraph_length	same_categorisation_raw	same_categorisation_percentage	lda_dominant_category
0	amsterdam_antwerp	604	28.750000	535	0.885762	lda_art
1	amsterdam_bilbao	37	23.513514	30	0.810811	lda_entertainment
2	amsterdam_bremen	81	28.395062	72	0.888889	lda_transportation
3	amsterdam_dortmund	47	25.723404	44	0.936170	lda_transportation
4	amsterdam_dresden	128	28.421875	113	0.882812	lda_entertainment
...	...	...	...	...	...	...
1621	zürich_bremen	23	29.478261	20	0.869565	lda_entertainment
1622	zürich_dresden	70	19.657143	63	0.900000	lda_entertainment
1623	zürich_oslo	52	22.000000	41	0.788462	lda_education
1624	zürich_palermo	13	28.000000	13	1.000000	lda_entertainment
1625	zürich_sheffield	5	58.000000	5	1.000000	lda_sport

embedding_dominant_category	lda_art	embedding_art	lda_diplomacy	embedding_diplomacy	lda_education	embedding_education	lda_entertainment
art	262	288	85	74	45	31	65
entertainment	1	5	4	4	3	2	18
art	13	19	15	10	10	8	16
transportation	3	2	3	4	2	2	9
entertainment	40	51	5	3	11	7	63
...	...	...	...	...	...	...	...
sport	0	1	1	1	2	1	9
entertainment	7	11	2	3	11	8	38
entertainment	2	9	2	4	19	10	12
entertainment	1	1	1	1	0	0	8
sport	0	0	0	0	0	0	1

embedding_entertainment	lda_sport	embedding_sport	lda_transportation	embedding_transportation
62	40	48	107	101
16	3	4	8	6
16	11	11	16	17
9	12	13	18	17
56	3	4	6	7
...	...	...	...	...
8	8	9	3	3
35	8	9	4	4
11	11	11	6	7
8	3	3	0	0
1	4	4	0	0

1626 rows × 19 columns

Figure 22: *Soft and hard Classification of the 1770 city pairs belonging to the 60 cities.*

Appendix B - Extra Information

B.1 SpaCy

TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	PROPN	NNP	nsubj	Xxxxx	True	False
is	be	AUX	VBZ	aux	xx	True	True
looking	look	VERB	VBG	ROOT	xxxx	True	False
at	at	ADP	IN	prep	xx	True	True
buying	buy	VERB	VBG	pcomp	xxxx	True	False
U.K.	u.k.	PROPN	NNP	compound	X.X.	False	False
startup	startup	NOUN	NN	dobj	xxxx	True	False
for	for	ADP	IN	prep	xxx	True	True
$	$	SYM	$	quantmod	$	False	False
1	1	NUM	CD	compound	d	False	False
billion	billion	NUM	CD	pobj	xxxx	True	False

Text: The original word text.
Lemma: The base form of the word.
POS: The simple part-of-speech tag.
Tag: The detailed part-of-speech tag.
Dep: Syntactic dependency, i.e. the relation between tokens.
Shape: The word shape -capitalization, punctuation, digits.
is alpha: Is the token an alpha character?
is stop: is the token part of a stop list. i.e. the most common words of the language?

Figure 23: *SpaCy possible attributes of a token.*

B.2 Topic Coherence Model

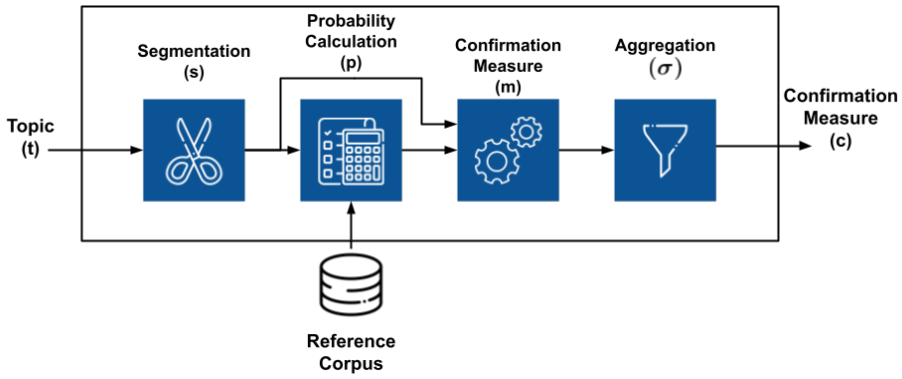


Figure 24: *Structure of the four stage topic coherence pipeline from the paper. [29].*

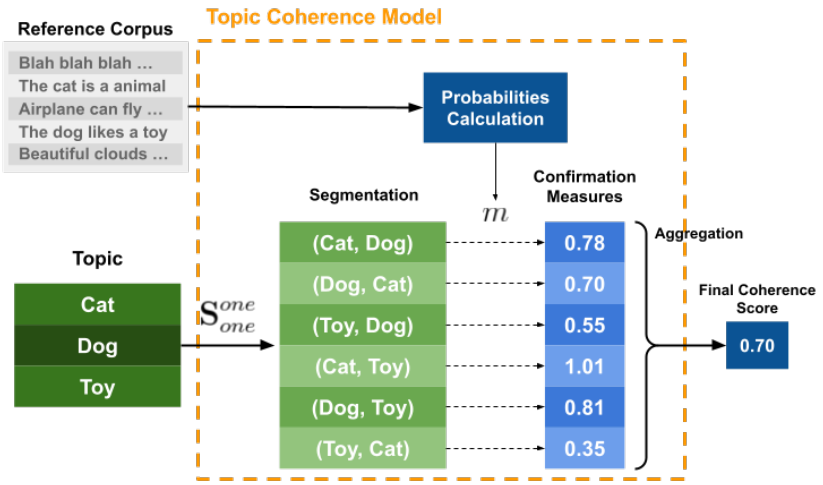


Figure 25: *Visualisation of the coherence calculation of an individual topic.*

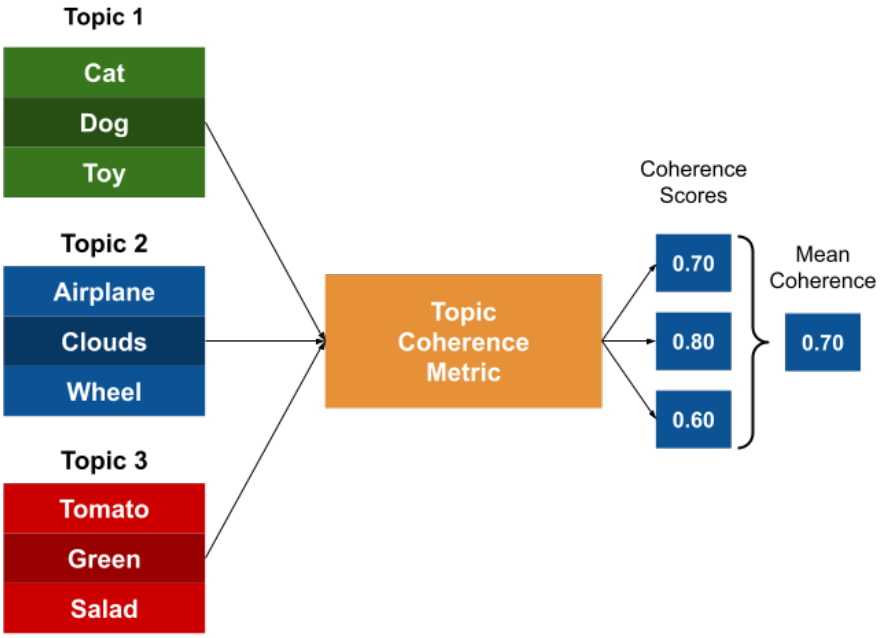


Figure 26: *Visualisation of the total coherence score of an LDA topic model based on its topics.*