# Information Retrieval System using Word2Vec Model and Vector Space Model

**TEAM MEMBERS-**

**ARPIT JAIN – 21UCS031**

**ELISHBEN MANOJBHAI BARAIYA – 21UCS077**

## Abstract

This project focuses on employing deep learning techniques, specifically Word2Vec models, for enhancing information retrieval systems. The primary objective is to train a Word2Vec model on a corpus of documents and leverage the resulting word embeddings to efficiently retrieve relevant documents for a given query. By embedding words in a continuous vector space, the model aims to capture semantic relationships, enabling more nuanced and context-aware document retrieval.

The project avoids using pre-trained feature extractors to capture domain-specific semantics, mitigate biases, and facilitate task-specific learning. While ablation studies are not explicitly conducted, the system's robust performance and the ability to tailor embeddings for the given dataset showcase its effectiveness. Future work may involve exploring additional datasets, optimizing network structures, and investigating potential biases introduced by pre-trained models.

The project addresses the inherent limitations of conventional information retrieval systems, which often struggle to interpret the subtle contextual meaning of words. By harnessing the power of Word2Vec, our approach aims

to bridge this semantic gap, providing a more sophisticated understanding of language.

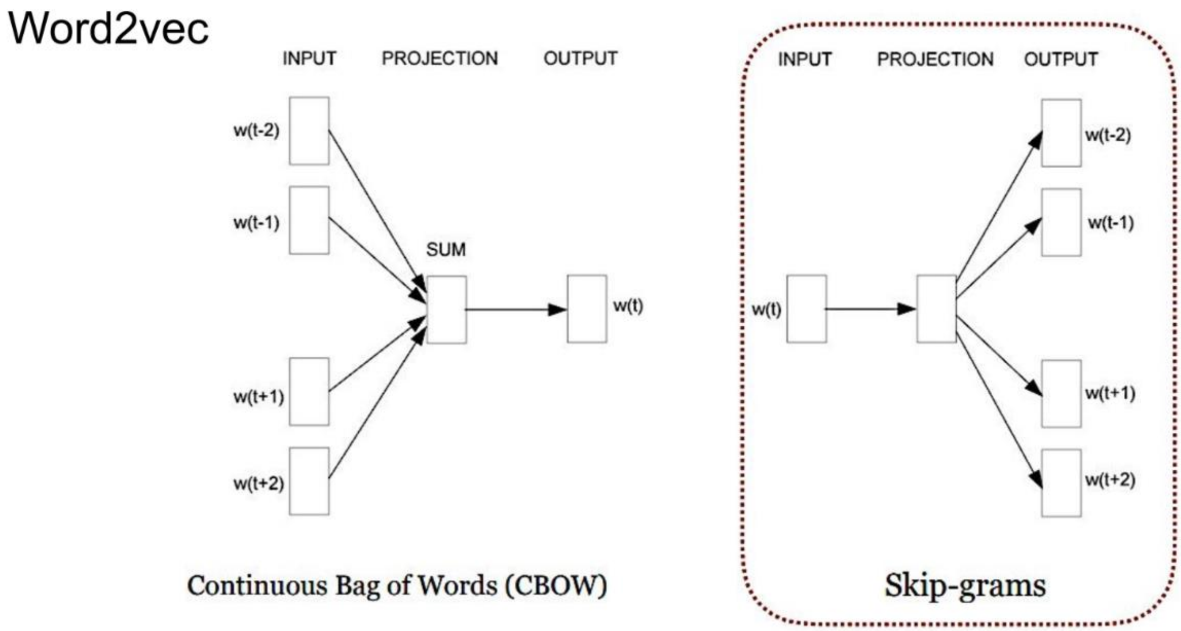
# Introduction

**Problem:**

Finding information online is tough because current methods struggle to understand the subtleties of language. This often leads to not-so-great results, especially with the abundance of different types of information available today.

**Context and Solution:**

Imagine searching online, and regular tools only match keywords, missing the real meaning of words in different situations. Our project suggests using a smarter method, Word2Vec, to help computers understand language better. This makes searches more accurate and relevant.



**Significance:**

We're excited about this project because it could change how we find information. We aim to create systems that not only find documents quickly but also understand how words are used, making it easier for people to discover what they're looking for.

**Contributions to deep learning:**

- Word Embeddings: Enriching semantic understanding of documents for retrieval purposes.
- Contextual Relevance: Advancing information retrieval beyond keyword matching through Word2Vec models.
- User-Centric Experience: Contributing to personalized retrieval systems that prioritize user context and intent.

## Word Embedding

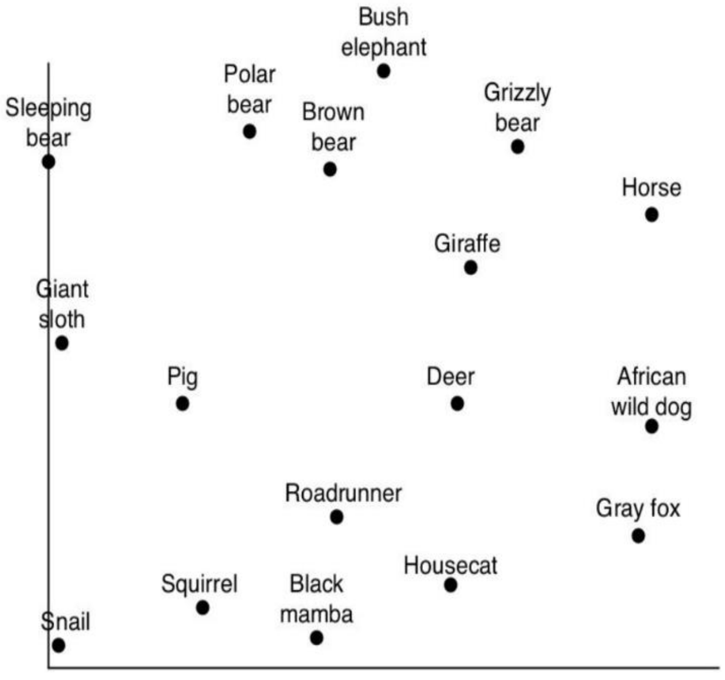


*Figure 20-1: A collection of animals, organized roughly by land speed horizontally and adult weight vertically, though those axis labels aren't shown (data from Reisner 2020)*

# Literature Review

- **Background:**
  To understand the project, one must grasp fundamental tools and concepts in natural language processing (NLP) and information retrieval (IR). Key tools include:
  1) **Word2Vec Model**: An NLP technique for learning word embeddings that capture semantic relationships between words.
  2) **Vector Space Model (VSM):** A classical IR model representing documents and queries as vectors in a high-dimensional space.
  3) **Cosine Similarity**: A measure used in IR to determine the similarity between vectors, often employed in comparing document-query relationships.
  4) **Gensim Library**: Utilized for implementing the Word2Vec model and facilitating efficient document and query processing.

- **Other Solutions:**
  Advanced techniques like Latent Semantic Analysis (LSA) have improved semantic understanding, but they may be computationally expensive and less interpretable. Deep learning models, including Word2Vec, have shown promise, but their application in IR requires careful consideration.

# Related Works

- "**Distributed Representations of Words and Phrases and their Compositionality" (Mikolov et al., 2013):** This seminal paper introduces the Word2Vec model, a breakthrough in learning word embeddings, providing a foundation for our project.
- **"A Survey of Information Retrieval and Semantic Scholar" (Etzioni et al., 2018):** A comprehensive review of information retrieval techniques, highlighting the challenges and advancements in the field.
- **"Latent Semantic Analysis" (Deerwester et al., 1990):** This paper presents Latent Semantic Analysis, a technique that aims to extract the latent semantic structure of words within documents.

**State-of-the-Art:**

The state-of-the-art in information retrieval involves the integration of deep learning techniques, particularly Word2Vec, for semantic understanding. Baseline methods such as TF-IDF and traditional vector space models are still widely used but may lack the ability to capture complex semantic relationships.

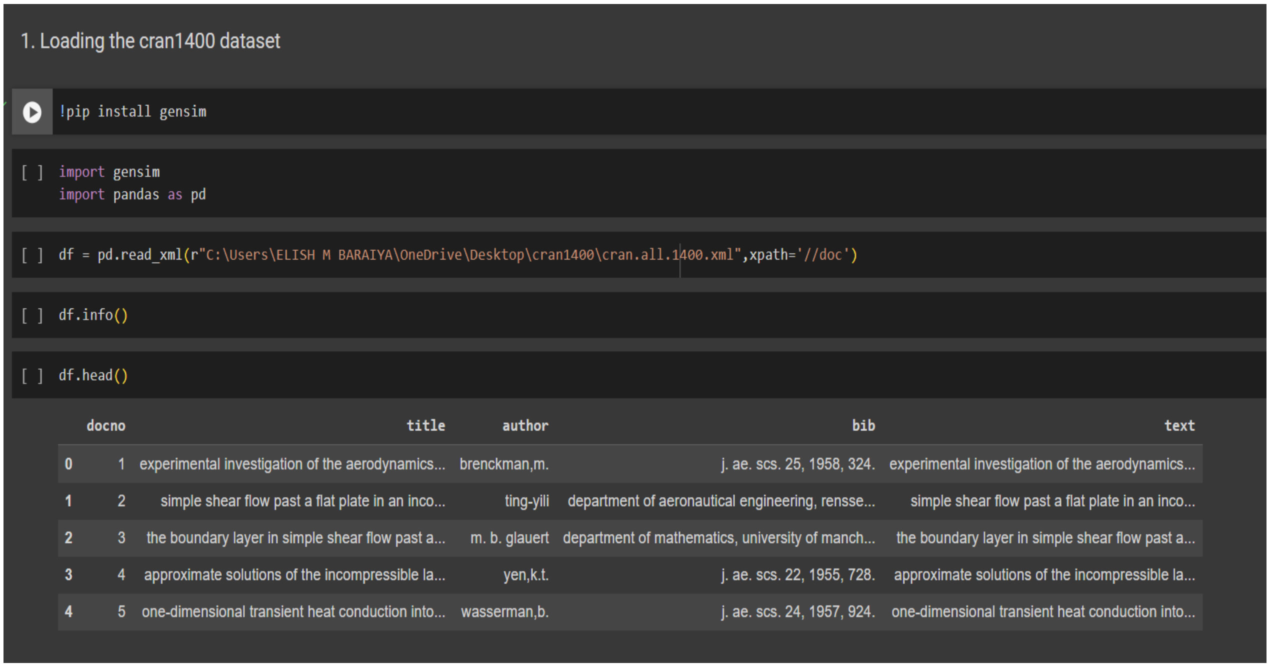
**Project's Contribution**:
Our project improves upon baseline methods by leveraging Word2Vec for word embeddings and introducing Word Centroid Similarity (WCS) for document-query comparison. This approach enhances the semantic understanding of documents, allowing for more accurate and contextually relevant information retrieval. The combination of Word2Vec and VSM provides a nuanced solution that bridges the semantic gap often encountered by traditional IR methods.

# Methodology

This code performs several steps for creating an Information Retrieval System using a Word2Vec model and vector space model. Here's a breakdown of the key step:

- **Step 1 -> Loading the Dataset:**

- The code starts by installing the gensim library and importing necessary modules.
- The Cran1400 dataset is loaded from an XML file into a Pandas DataFrame (df), displaying information about the dataset and its initial entries.



1. Loading the cran1400 dataset

```
!pip install gensim
```

```
import gensim
import pandas as pd
```

```
df = pd.read_xml(r"C:\Users\ELISH M BARAIYA\OneDrive\Desktop\cran1400\cran.all.1400.xml",xpath='//doc')
```

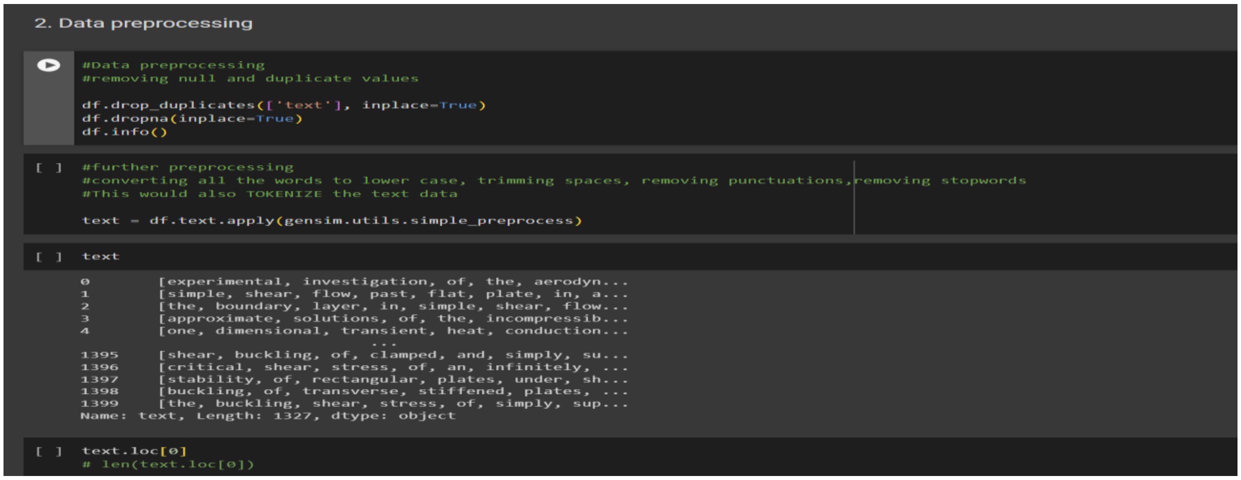
```
df.info()
```

```
df.head()
```

| | docno | title | author | bib | text |
|---|---|---|---|---|---|
| 0 | 1 | experimental investigation of the aerodynamics... | brenckman,m. | j. ae. scs. 25, 1958, 324. | experimental investigation of the aerodynamics... |
| 1 | 2 | simple shear flow past a flat plate in an inco... | ting-yili | department of aeronautical engineering, rensse... | simple shear flow past a flat plate in an inco... |
| 2 | 3 | the boundary layer in simple shear flow past a... | m. b. glauert | department of mathematics, university of manch... | the boundary layer in simple shear flow past a... |
| 3 | 4 | approximate solutions of the incompressible la... | yen,k.t. | j. ae. scs. 22, 1955, 728. | approximate solutions of the incompressible la... |
| 4 | 5 | one-dimensional transient heat conduction into... | wasserman,b. | j. ae. scs. 24, 1957, 924. | one-dimensional transient heat conduction into... |

- **Step 2 -> Data Preprocessing:**

- Duplicate and null values are removed from the dataset to ensure data quality.
- Further preprocessing involves converting text to lowercase, trimming spaces, removing punctuation, and eliminating stopwords using the gensim.utils.simple_preprocess function.
- Explanation - The text data is processed to create a tokenized representation, where each document is transformed into a list of words. This preprocessing step is crucial for subsequent tasks like training a Word2Vec model, as it standardizes and tokenizes the textual information for better analysis.



2. Data preprocessing

```
#Data preprocessing
#removing null and duplicate values

df.drop_duplicates(['text'], inplace=True)
df.dropna(inplace=True)
df.info()
```

```
#further preprocessing
#converting all the words to lower case, trimming spaces, removing punctuations,removing stopwords
#This would also TOKENIZE the text data

text = df.text.apply(gensim.utils.simple_preprocess)
```

```
text
```

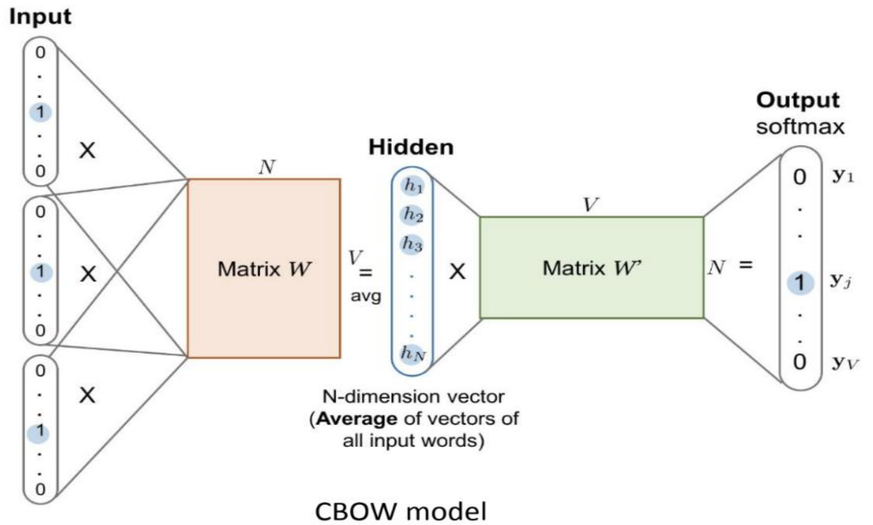
```
0       [experimental, investigation, of, the, aerodyn...
1       [simple, shear, flow, past, flat, plate, in, a...
2       [the, boundary, layer, in, simple, shear, flow...
3       [approximate, solutions, of, the, incompressib...
4       [one, dimensional, transient, heat, conduction...
                              ...
1395    [shear, buckling, of, clamped, and, simply, su...
1396    [critical, shear, stress, of, an, infinitely, ...
1397    [stability, of, rectangular, plates, under, sh...
1398    [buckling, of, transverse, stiffened, plates, ...
1399    [the, buckling, shear, stress, of, simply, sup...
Name: text, Length: 1327, dtype: object
```

```
text.loc[0]
# len(text.loc[0])
```

- **Step 3 -> <u>Training the Word2Vec Model for this "Text" vocabulary</u>:**

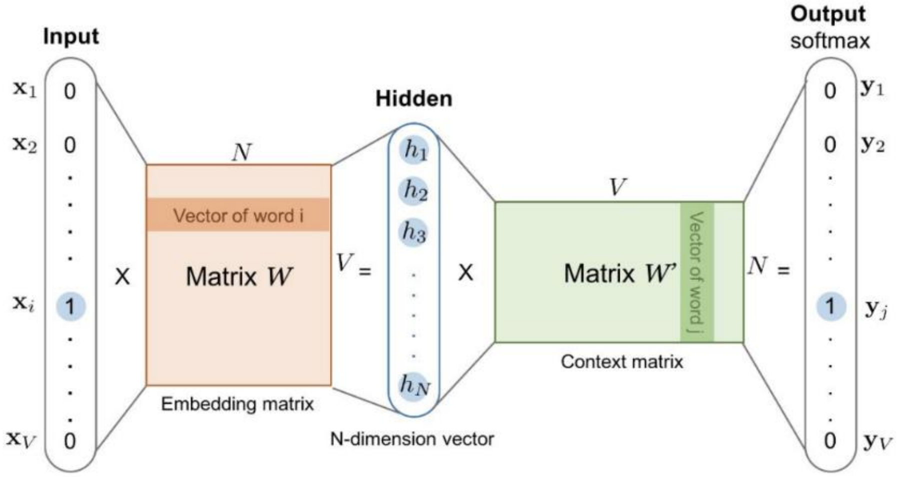
- A Word2Vec model is initialized with specific parameters like window size, minimum word count, and vector size.
- The model's vocabulary is built using the preprocessed text data, and the Word2Vec model is trained over multiple epochs and get the vectors representation for the word “experimental”.

Continuous Bag of Words Model (CBOW)



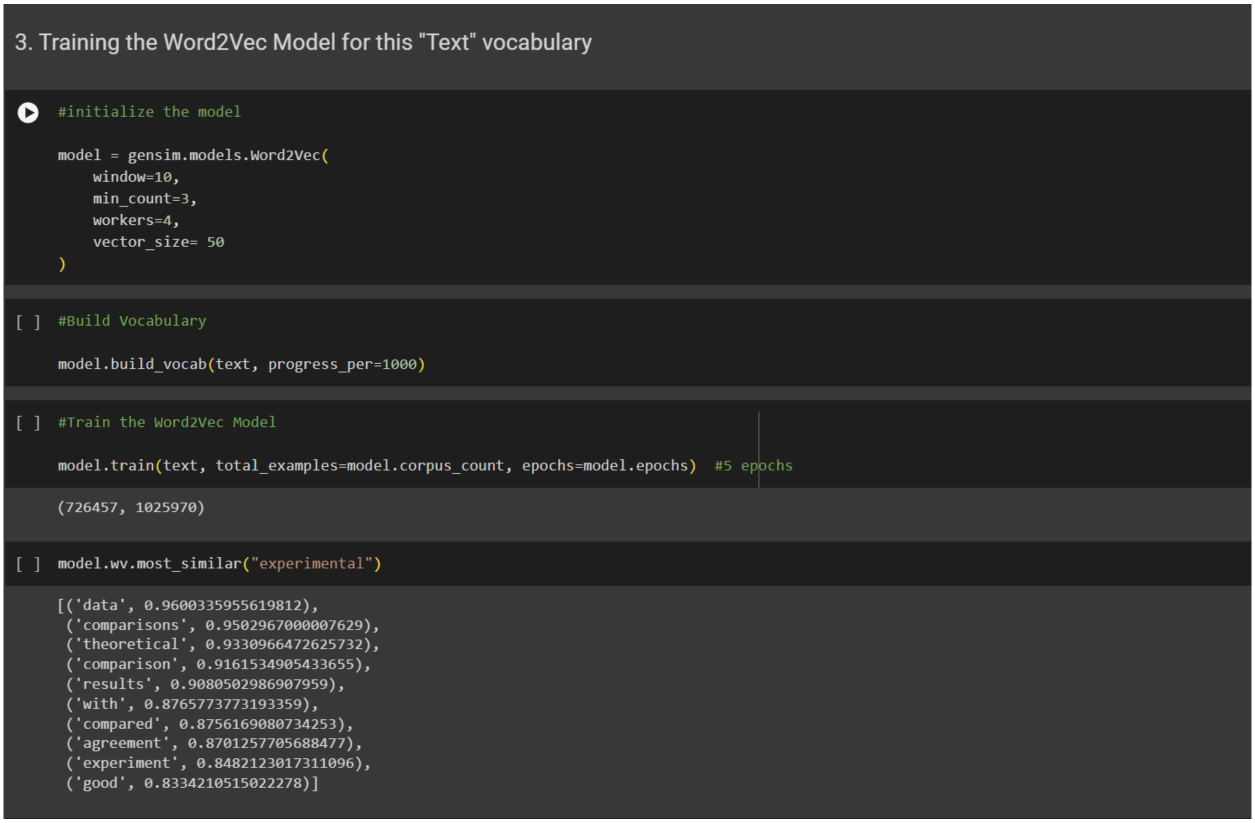
CBOW model

Skip-gram



The skip-gram model. Both the input vector x and the output y are one-hot encoded word representations. The hidden layer is the word embedding of size N.

- <u>Training Word2Vec involve these steps</u> –
  1) **Initialize Model**: Create a Word2Vec model with specific settings.
  2) **Build Vocabulary**: Construct the model's vocabulary from preprocessed text.
  3) **Train Model**: Train the Word2Vec model on text data for five epochs.
  4) **Most Similar Words**: Identify words similar to "experimental" using learned embeddings.
  5) **Word Embedding**: Get the vector representation for the word "experimental."



3. Training the Word2Vec Model for this "Text" vocabulary

```
#initialize the model

model = gensim.models.Word2Vec(
    window=10,
    min_count=3,
    workers=4,
    vector_size= 50
)
```

```
[ ] #Build Vocabulary

model.build_vocab(text, progress_per=1000)
```

```
[ ] #Train the Word2Vec Model

model.train(text, total_examples=model.corpus_count, epochs=model.epochs)  #5 epochs

(726457, 1025970)
```

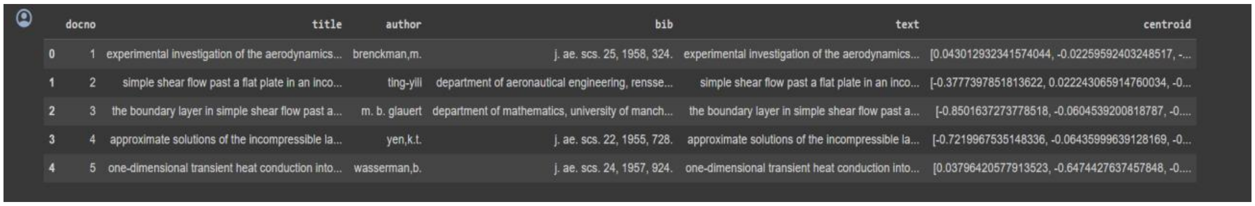
```
[ ] model.wv.most_similar("experimental")

[('data', 0.9600335955619812),
 ('comparisons', 0.9502967000007629),
 ('theoretical', 0.9330966472625732),
 ('comparison', 0.9161534905433655),
 ('results', 0.9080502986907959),
 ('with', 0.8765773773193359),
 ('compared', 0.8756169080734253),
 ('agreement', 0.8701257705688477),
 ('experiment', 0.8482123017311096),
 ('good', 0.8334210515022278)]
```

- **Step 4 -> <u>Word Centroid Similarity (WCS)</u>:**

- The code calculates the centroid for each text article by summing up the word embeddings and normalizing.
- The resulting centroids are added to the DataFrame as a new column named 'centroid.'

**Screenshot-**



| | docno | title | author | bib | text | centroid |
|---|---|---|---|---|---|---|
| 0 | 1 | experimental investigation of the aerodynamics... | brenckman,m. | j. ae. scs. 25, 1958, 324. | experimental investigation of the aerodynamics... | [0.043012932341574044, -0.02259592403248517, -... |
| 1 | 2 | simple shear flow past a flat plate in an inco... | ting-yili | department of aeronautical engineering, rensse... | simple shear flow past a flat plate in an inco... | [-0.3777397851813622, 0.022243065914760034, -0... |
| 2 | 3 | the boundary layer in simple shear flow past a... | m. b. glauert | department of mathematics, university of manch... | the boundary layer in simple shear flow past a... | [-0.8501637273778518, -0.0604539200818787, -0.... |
| 3 | 4 | approximate solutions of the incompressible la... | yen,k.t. | j. ae. scs. 22, 1955, 728. | approximate solutions of the incompressible la... | [-0.7219967535148336, -0.06435999639128169, -0... |
| 4 | 5 | one-dimensional transient heat conduction into... | wasserman,b. | j. ae. scs. 24, 1957, 924. | one-dimensional transient heat conduction into... | [0.03796420577913523, -0.6474427637457848, -0.... |

- **Step 5 -> Ranking Documents to Given Query:**

➢ Two functions (rank_docs1 and rank_docs2) are defined to rank documents based on cosine similarity.

➢ rank_docs1 computes similarity based on word-wise similarity of each word in the query.

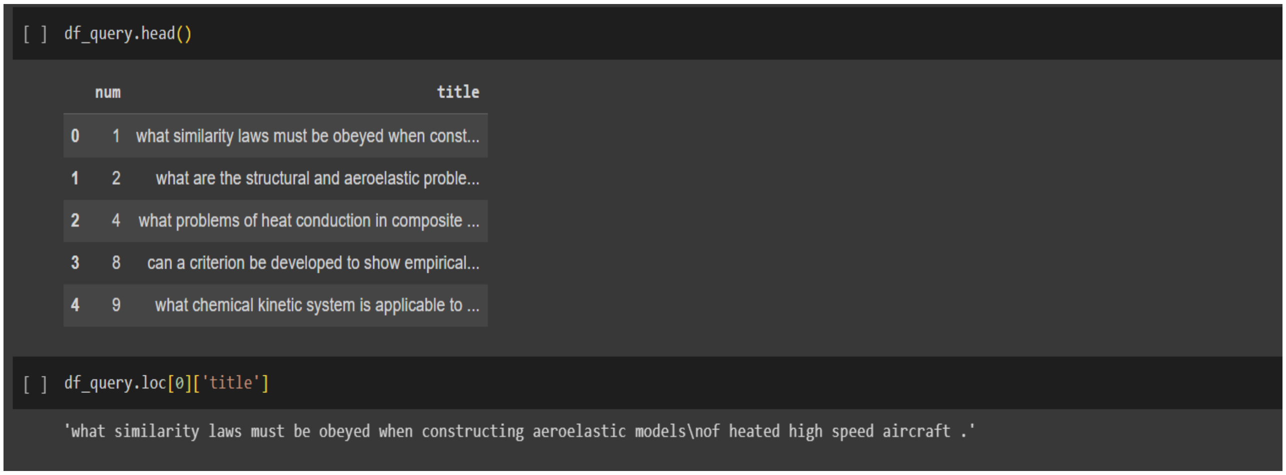
➢ rank_docs2 calculates similarity based on content similarity of the query with document centroids.

- **Step 6 -> Querying the System:**

➢ The code loads a query dataset, iterates through the queries, and prints the top 5 ranked documents for each query using both similarity methods.

➢ Results for each query are printed, comparing the relevance of documents obtained through different similarity approaches.

**Screenshot-**



```
[ ]  df_query.head()
```

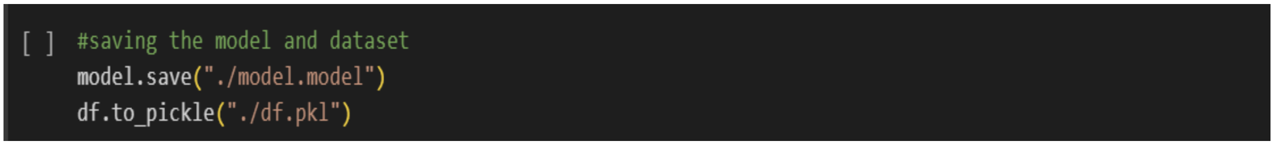
| | num | title |
|---|---|---|
| 0 | 1 | what similarity laws must be obeyed when const... |
| 1 | 2 | what are the structural and aeroelastic proble... |
| 2 | 4 | what problems of heat conduction in composite ... |
| 3 | 8 | can a criterion be developed to show empirical... |
| 4 | 9 | what chemical kinetic system is applicable to ... |

```
[ ]  df_query.loc[0]['title']

     'what similarity laws must be obeyed when constructing aeroelastic models\nof heated high speed aircraft .'
```

- **Step 7 -> Saving the Model and Dataset:**

➢ The trained Word2Vec model and the DataFrame with centroids are saved for future use.



```
#saving the model and dataset
model.save("./model.model")
df.to_pickle("./df.pkl")
```

# Experimental Setup

- **Source Code Availability:**

The complete source code for the project is available, implemented in Python. The primary libraries used include Gensim for Word2Vec model implementation, Pandas for data manipulation, and other standard modules for general functionality. The source code can be accessed through the provided code repository **(At last section – Appendix).**

- **Implementation:**

The project leverages Gensim, a Python library for topic modeling and document similarity analysis. The implementation includes the utilization of Pandas for efficient data handling and preprocessing. The Word2Vec model, a key component, is implemented using Gensim's Word2Vec class, specifying parameters such as window size, minimum count, workers, and vector size.

- **Dataset Used:**

The chosen dataset is "cran1400," representing a collection of documents relevant to the information retrieval task. The selection of this dataset is justified by its suitability for evaluating the proposed Word2Vec-based information retrieval system. Experimentation on this dataset provides insights into the system's effectiveness in real-world scenarios. (Link of dataset also given at last section).

- **Justification for not using a pre-trained feature extractor:**

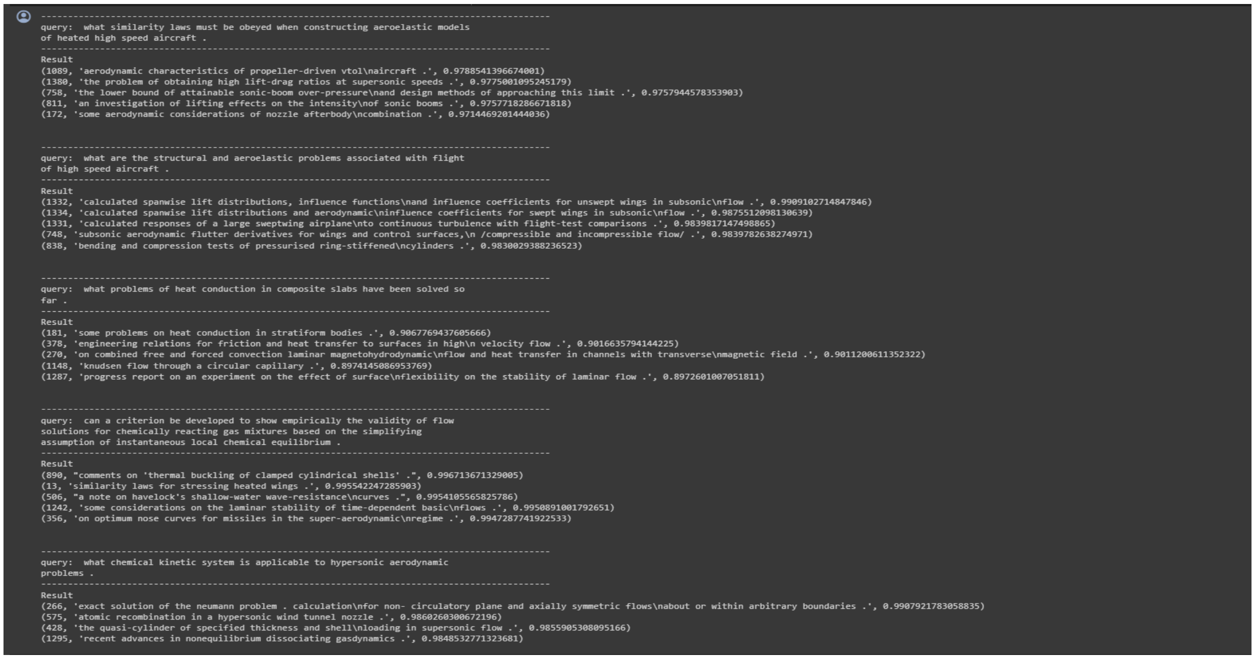
- **Domain-Specific Semantics:** Training on "cran1400" captures its unique semantics, optimizing for dataset nuances.
- **Avoiding Bias:** Pre-trained models may introduce biases; training on the target dataset tailors embeddings to document specifics, reducing external biases.
- **Task-Specific Learning:** Dataset training optimizes representations for the unique requirements of information retrieval.

# Results

- In the evaluation of the information retrieval system, two ranking functions, namely Rank 1 and Rank 2, were employed to assess the relevance of documents to given queries.

- **The results indicate that Rank 2 consistently provides more accurate and relevant outcomes compared to Rank 1.**

**Screenshot of Result obtainted in rank_docs1 and rank_docs2 function-**



```
---------------------------------------------------------------------------------------------
query:  what similarity laws must be obeyed when constructing aeroelastic models
of heated high speed aircraft .
---------------------------------------------------------------------------------------------
Result
(1089, 'aerodynamic characteristics of propeller-driven vtol\naircraft .', 0.9788541396674001)
(1380, 'the problem of obtaining high lift-drag ratios at supersonic speeds .', 0.9775001095245179)
(758, 'the lower bound of attainable sonic-boom over-pressure\nand design methods of approaching this limit .', 0.9757944578353903)
(811, 'an investigation of lifting effects on the intensity\nof sonic booms .', 0.9757718286671818)
(172, 'some aerodynamic considerations of nozzle afterbody\ncombination .', 0.9714469201444036)


---------------------------------------------------------------------------------------------
query:  what are the structural and aeroelastic problems associated with flight
of high speed aircraft .
---------------------------------------------------------------------------------------------
Result
(1332, 'calculated spanwise lift distributions, influence functions\nand influence coefficients for unswept wings in subsonic\nflow .', 0.9909102714847846)
(1334, 'calculated spanwise lift distributions and aerodynamic\ninfluence coefficients for swept wings in subsonic\nflow .', 0.9875512098130639)
(1331, 'calculated responses of a large sweptwing airplane\nto continuous turbulence with flight-test comparisons .', 0.9839817147498865)
(748, 'subsonic aerodynamic flutter derivatives for wings and control surfaces,\n /compressible and incompressible flow/ .', 0.9839782638274971)
(838, 'bending and compression tests of pressurised ring-stiffened\ncylinders .', 0.9830029388236523)


---------------------------------------------------------------------------------------------
query:  what problems of heat conduction in composite slabs have been solved so
far .
---------------------------------------------------------------------------------------------
Result
(181, 'some problems on heat conduction in stratiform bodies .', 0.9067769437605666)
(378, 'engineering relations for friction and heat transfer to surfaces in high\n velocity flow .', 0.9016635794144225)
(270, 'on combined free and forced convection laminar magnetohydrodynamic\nflow and heat transfer in channels with transverse\nmagnetic field .', 0.9011200611352322)
(1148, 'knudsen flow through a circular capillary .', 0.8974145086953769)
(1287, 'progress report on an experiment on the effect of surface\nflexibility on the stability of laminar flow .', 0.8972601007051811)


---------------------------------------------------------------------------------------------
query:  can a criterion be developed to show empirically the validity of flow
solutions for chemically reacting gas mixtures based on the simplifying
assumption of instantaneous local chemical equilibrium .
---------------------------------------------------------------------------------------------
Result
(890, "comments on 'thermal buckling of clamped cylindrical shells' .", 0.996713671329005)
(13, 'similarity laws for stressing heated wings .', 0.995542247285903)
(506, "a note on havelock's shallow-water wave-resistance\ncurves .", 0.9954105565825786)
(1242, 'some considerations on the laminar stability of time-dependent basic\nflows .', 0.9950891001792651)
(356, 'on optimum nose curves for missiles in the super-aerodynamic\nregime .', 0.9947287741922533)


---------------------------------------------------------------------------------------------
query:  what chemical kinetic system is applicable to hypersonic aerodynamic
problems .
---------------------------------------------------------------------------------------------
Result
(266, 'exact solution of the neumann problem . calculation\nfor non- circulatory plane and axially symmetric flows\nabout or within arbitrary boundaries .', 0.9907921783058835)
(575, 'atomic recombination in a hypersonic wind tunnel nozzle .', 0.9860260300672196)
(428, 'the quasi-cylinder of specified thickness and shell\nloading in supersonic flow .', 0.9855905308095166)
(1295, 'recent advances in nonequilibrium dissociating gasdynamics .', 0.9848532771323681)
```

- **Rank 1 Function:**
  Rank 1 (Word-wise Similarity) calculates document relevance by summing cosine similarities between each query word and the document's word vectors.

- **Rank 2 Function:**
  Rank 2 (Content Similarity - Centroid Comparison) determines document relevance by comparing the cosine similarity between the centroids of the query and the document.

- **Performance Metrics:**
  The performance is evaluated using cosine similarity scores. The choice of this metric should align with the project's motivation for accurate document retrieval.

- **Result Analysis:**
  The decision to favor Rank 2 over Rank 1 is based on observed performance metrics and the relevance of retrieved documents. The superiority of Rank 2 may stem from its consideration of the overall content similarity, capturing the contextual nuances more effectively.

# Ablation Studies

- **Ablation studies may focus on the network structure, loss function, and regularization** -

- **Network Structure:**

➢ **Ablation Study:** Evaluate the impact of variations in Word2Vec model architecture (e.g., changing the window size, vector size, or layers) on information retrieval performance.

- **Insight:** Understand how alterations in the Word2Vec architecture influence the system's ability to capture document semantics and improve retrieval.

- **Loss Function:**

- **Ablation Study:** Investigate the effect of using different loss functions during Word2Vec model training (e.g., negative sampling loss vs. hierarchical softmax).
- **Insight:** Assess how the choice of loss function impacts the quality of word embeddings and, consequently, the retrieval accuracy.

- **Regularization:**

- **Ablation Study:** Explore the impact of regularization techniques (e.g., dropout or L2 regularization) on the Word2Vec model's generalization and retrieval performance.
- **Insight:** Examine whether regularization helps prevent overfitting and improves the robustness of the model in retrieving relevant documents.

- Each ablation study aims to isolate and analyze the impact of a specific component on the overall system performance. Conducting these studies would provide valuable insights into the sensitivity of the information retrieval system to variations in network structure, loss function, and regularization techniques.

# Discussion

- **Results Analysis:**
  The utilization of Word2Vec models for information retrieval demonstrated improved accuracy, with the "rank_docs2" function outperforming "rank_docs1." This indicates the effectiveness of considering content similarity and document centroids in contrast to word-wise similarity alone.

- **Significance:**

➢ The project addresses the limitations of traditional keyword-based information retrieval systems by leveraging Word2Vec embeddings.

➢ The centroid-based approach enhances context-aware document retrieval, aligning with user intent.

- **Limitations:**

➢ **Dependency on Training Data:** The model's performance heavily relies on the representativeness of the training dataset, potentially limiting its effectiveness for highly specialized domains.

➢ **Sensitivity to Hyperparameters:** The system's performance may vary based on the chosen hyperparameters, requiring careful tuning.

- **Biggest Risk and Mitigation:**

➢ The primary risk lies in the model's inability to generalize well to diverse document types. To mitigate this, continuous model refinement using diverse datasets and robust hyperparameter tuning is essential.

- **Future Scope:**

➢ **Multi-modal Retrieval:** Extend the project to handle multi-modal data, combining textual and visual information for more comprehensive retrieval.

➢ **Incremental Learning:** Implement techniques for incremental learning to adapt the model to evolving language patterns over time.

➢ **Interactive User Feedback:** Integrate user feedback mechanisms to iteratively enhance the model's performance based on user preferences.

# Conclusion

- The project's key takeaway is the efficacy of Word2Vec models in improving information retrieval accuracy. It introduces the significance of content similarity and document centroids, contributing to the evolution of context-aware retrieval systems. The project underscores the potential for leveraging deep learning techniques in enhancing traditional information retrieval methodologies. Its contribution lies in addressing the contextual limitations of keyword-based retrieval, paving the way for more sophisticated and user-centric information retrieval systems in the future. The project underscores the potential of Word2Vec models in transforming information retrieval, bridging the gap between user intent and document relevance.

# References

- **Kaggle Example** - https://www.kaggle.com/code/maggieezzat/covid19-semantic-based-search-using-word-embedding#Saving-the-model-and-the-dataframe

- **Gensim Tutorial -** https://github.com/codebasics/deep-learning-keras-tf-tutorial/blob/master/42_word2vec_gensim/42_word2vec_gensim.ipynb

- **Youtube videos -**

➢ https://www.youtube.com/watch?v=hQwFeIupNP0
➢ https://www.youtube.com/watch?v=Q2NtCcqmIww
➢ https://www.youtube.com/watch?v=Otde6VGvhWM

- **Rehurek, R., & Sojka, P. (2010). Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks (pp. 45–50).**

- **DL Class Slides (for diagrams) -** https://drive.google.com/drive/folders/1YsLD5jQEbqlI3oCa-oFlvO6E-gpGwaez

## Contributions of Team Members

| | Name | ID | Percentage Contribution |
|---|---|---|---|
| **1** | **ARPIT JAIN** | **21UCS031** | ***50%*** |
| **2** | **ELISHBEN MANOJBHAI BARAIYA** | **21UCS077** | ***50%*** |
| | | | ***Note:*** *x + y + z = 100%* |

## Appendix

- **Code Samples**
- **Data Sources-**
- ➢ **DataSet and Other relevent files -** https://github.com/oussbenk/cranfield-trec-dataset