Predators becoming the prey:

using statistical machine learning and computational linguistics to detect sexual predators

By Norbert Eke

Supervisor: Abdallah Mohamed

Outline

1.) **Introduction to the Problem**
2.) Research Questions and Objectives
3.) Introduction to Data Being Used
4.) Algorithm Proposed
5.) Results & Model Selection Explained
6.) Future Works and Conclusion

Background

- **One in five U.S. teenagers** who regularly use the Internet have received an unwanted sexual solicitation via the web. (Crimes Against Children Research Center)

Background

- **One in five U.S. teenagers** who regularly use the Internet have received an unwanted sexual solicitation via the web. (Crimes Against Children Research Center)

- Significant **increase in the number of aggressive** sexual predators present online (Wolak et al., 2008)

Background

- **One in five U.S. teenagers** who regularly use the Internet have received an unwanted sexual solicitation via the web. (Crimes Against Children Research Center)

- Significant **increase in the number of aggressive** sexual predators present online (Wolak et al., 2008)

- “Most **first encounters** between offenders and victims (76%) **happened in online chat rooms**” (Wolak et al., 2004)

Background

- **One in five U.S. teenagers** who regularly use the Internet have received an unwanted sexual solicitation via the web. (Crimes Against Children Research Center)

- Significant **increase in the number of aggressive** sexual predators present online (Wolak et al., 2008)

- "Most **first encounters** between offenders and victims (76%) **happened in online chat rooms**" (Wolak et al., 2004)

- There is a need for better **intelligent systems** that are capable of accurately **detecting sexual predator's dangerous behavior** online

Outline

1.) Problem Background
2.) **Research Questions and Objectives**
3.) Introduction to Data Being Used
4.) Algorithm Proposed
5.) Results & Model Selection Explained
6.) Future Works and Conclusion

Research Questions

1. How can modern computational linguistics **interpret** online chat room conversations?

Research Questions

1. How can modern computational linguistics **interpret** online chat room conversations?

2. How to **extract semantic details** from conversations, and **detect** conversations containing **malicious intent**?

Research Questions

1. How can modern computational linguistics **interpret** online chat room conversations?

2. How to **extract semantic details** from conversations, and **detect** conversations containing **malicious intent**?

3. **Which machine learning models can predict** whether or not a conversation contains sexual predatory behavior?

Objective/Mission Statement

Join the powers of computational linguistics with statistical machine learning and **design an approach** that can **detect and classify textual data** as containing sexual predatory or non-predatory behaviour.

Outline

1.) Problem Background
2.) Research Questions and Objectives
3.) **Introduction to Data Being Used**
4.) Algorithm Proposed
5.) Results & Model Selection Explained
6.) Future Works and Conclusion

Sexual Predator Identification task

- **Data obtained from PAN**, a community of experts on digital text forensics

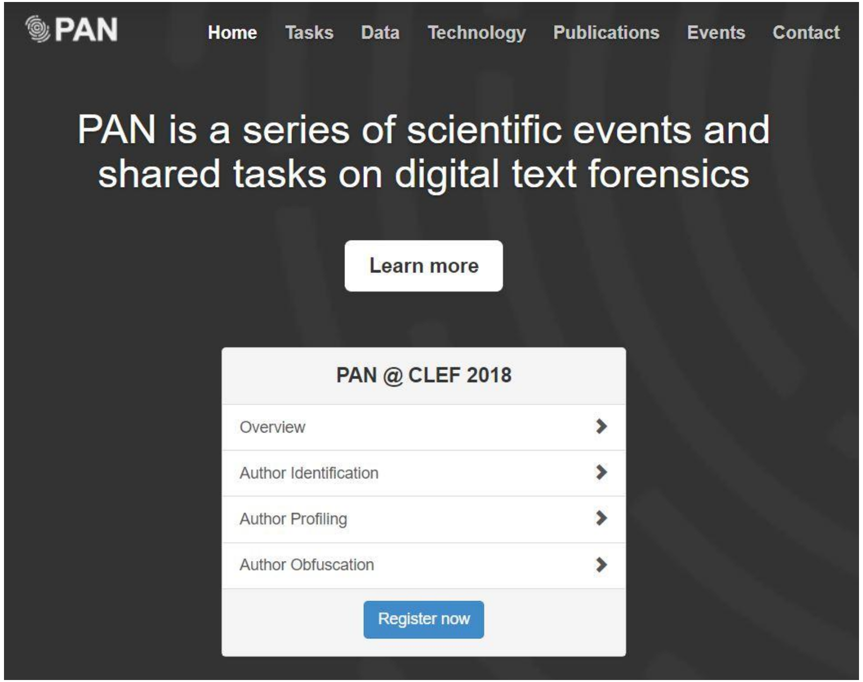


Figure 1: Website for PAN, a community of experts on digital text forensics

Sexual Predator Identification task

- **Data obtained from PAN**, a community of experts on digital text forensics

- Contains **thousands of labelled online chat logs**, where minors and adults pretending to be minors are chatting

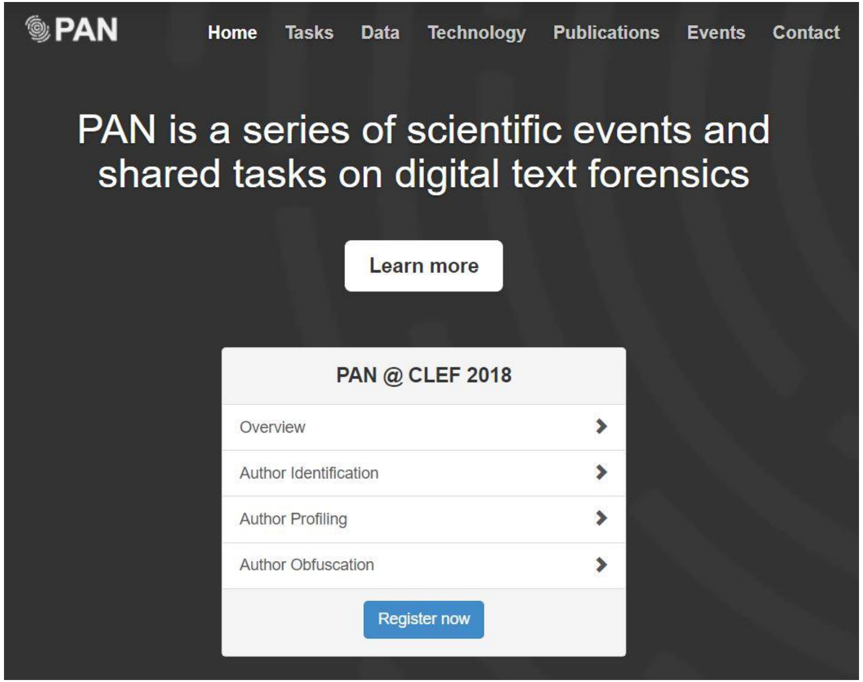


Figure 1: Website for PAN, a community of experts on digital text forensics

Sexual Predator Identification task

- **Data obtained from PAN**, a community of experts on digital text forensics

- Contains **thousands of labelled online chat logs**, where minors and adults pretending to be minors are chatting

- **Attributes include:** author id, conversation id, message line number, time, message content

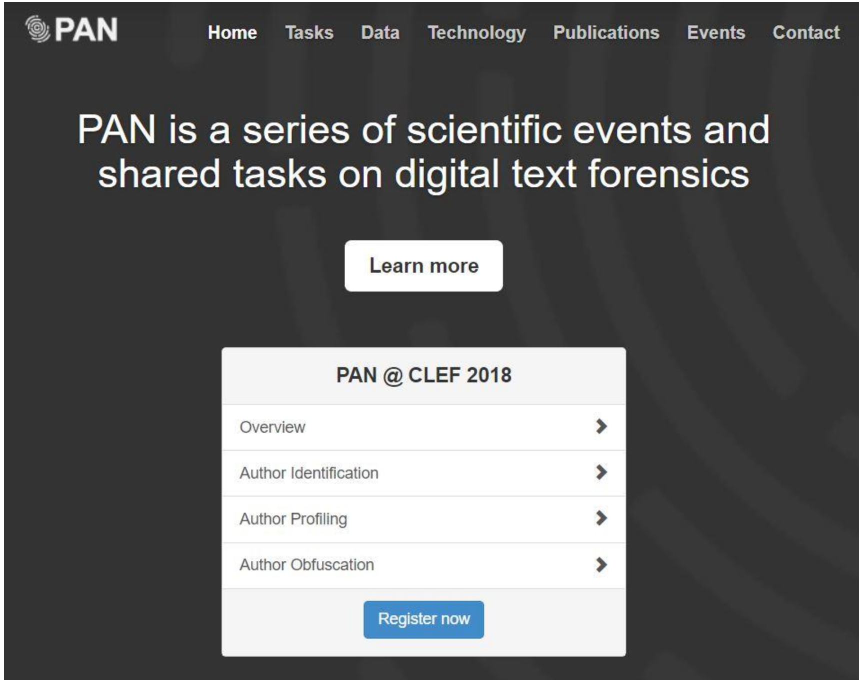


Figure 1: Website for PAN, a community of experts on digital text forensics

Sexual Predator Identification Data

Online chat-room conversations could contain:

- Misspelled Words
- Slang
- Internet Acronyms
- Inappropriate Language
- Broken Grammar
- Short, Messy and Unstructured Textual Data

Sexual Predator Identification Data

Online chat-room conversations could contain:

- Misspelled Words
- Slang
- Internet Acronyms
- Inappropriate Language
- Broken Grammar
- Short, Messy and Unstructured Textual Data


```
...
<conversation id="0042762e26ed295a8576806f5548cad9">
  <message line="3">
    <author>f069dbec9ab3e090972d432db279e3eb</author>
    <time>03:20</time>
    <text>whats up?</text>
  </message>
  <message line="4">
    <author>f069dbec9ab3e090972d432db279e3eb</author>
    <time>03:21</time>
    <text>how u doing?</text>
  </message>
  ...
  <message line="10">
    <author>f069dbec9ab3e090972d432db279e3eb</author>
    <time>04:00</time>
    <text>sse you llater?</text>
  </message>
</conversation>
...
<conversation id="0209b0a30c8eced86863631ada73a530">
  <message line="3">
    <author>0042762e26ed295a8576806f5548cad9</author>
    <time>01:17</time>
    <text>and that i dont touch u</text>
  </message>
</conversation>
```


Figure 2: Sample Raw Conversation data

Outline

1.) Problem Background
2.) Research Questions and Objectives
3.) Introduction to Data Being Used
4.) **Algorithm Proposed**
5.) Results & Model Selection Explained
6.) Future Works and Conclusion

Text Cleaning

- **Removal** of:
  - Extra white-spaces
  - HTML tag
  - Hyperlinks
  - Numeric characters
- Autocorrect - **spelling corrector**
- Lowercase conversion (for all words)
- Discarded conversations shorter than 3 words

```
>>> from autocorrect import spell
>>> spell('HTe')
'The'
```

Figure 3: Spelling corrector example

Word2Vec Deep Learning Model

- One of Google's **most famous** deep learning language models

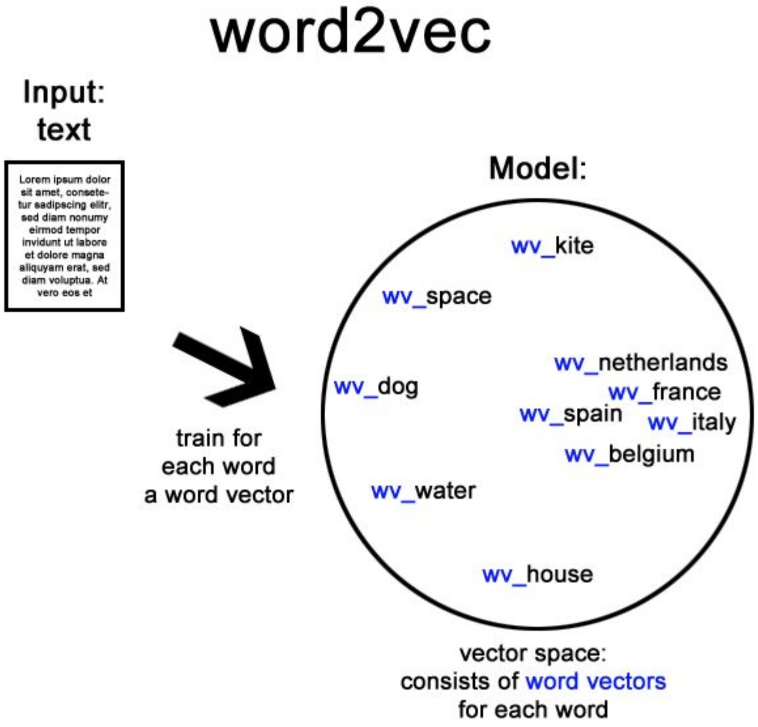


Figure 4: Visual Representation of Word2Vec Model

Word2Vec Deep Learning Model

- One of Google's **most famous** deep learning language models
- Model goes through **unsupervised learning** by getting trained on unlabelled textual data

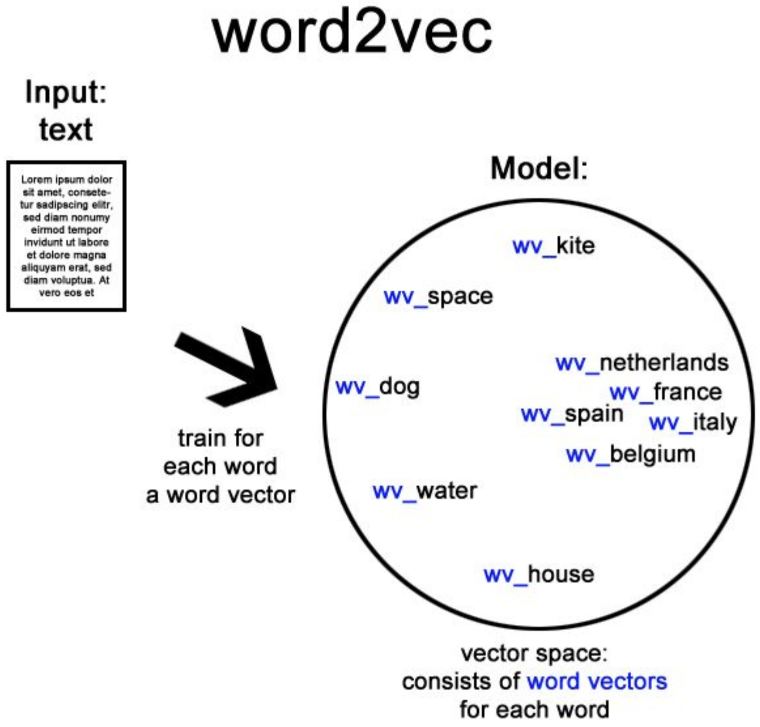


Figure 4: Visual Representation of Word2Vec Model

Word2Vec Deep Learning Model

- One of Google's **most famous** deep learning language models
- Model goes through **unsupervised learning** by getting trained on unlabelled textual data
- Word2Vec produces high dimensional vector representations of words (**word vectors**) as output

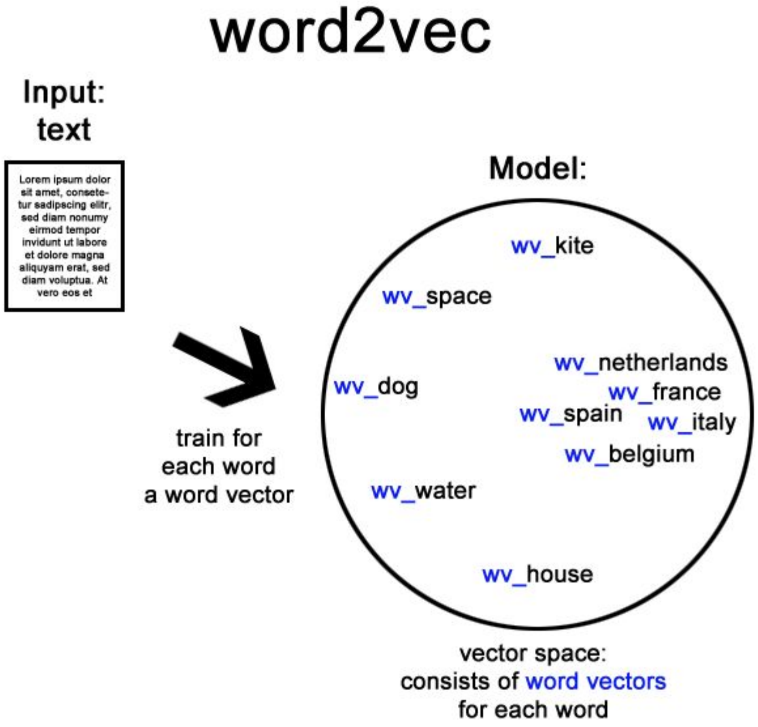


Figure 4: Visual Representation of Word2Vec Model

Word Vectors

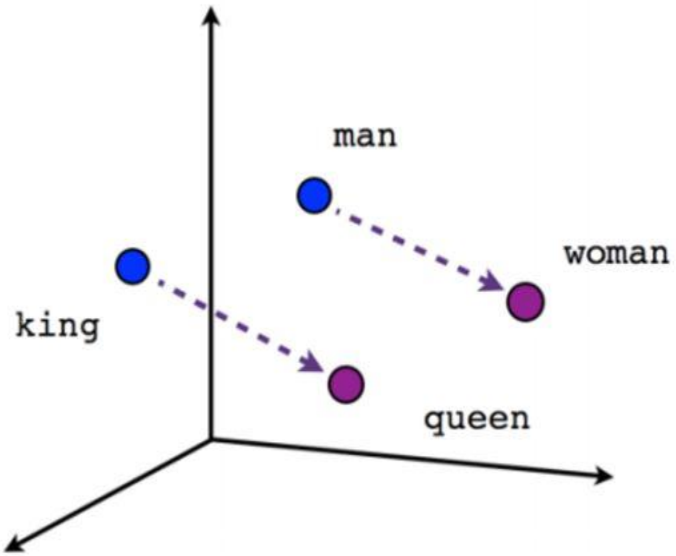


Figure 5: Male - Female Relationship visualized in a low dimensional vector space

- High dimensional **vector representation** of each word

Word Vectors

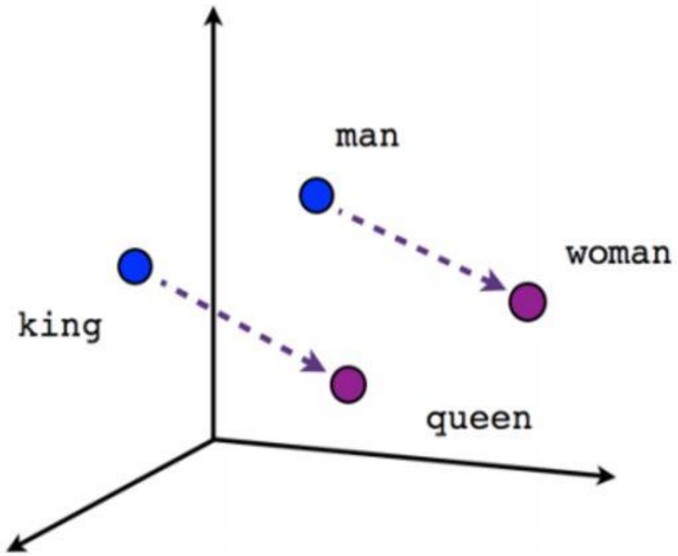


Figure 5: Male - Female Relationship visualized in a low dimensional vector space

- High dimensional **vector representation** of each word
- Used to **reconstruct linguistic context** of words

Word Vectors

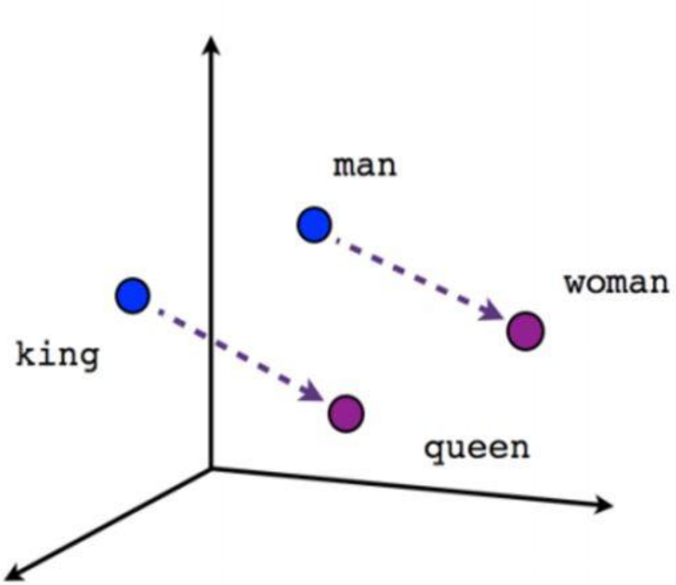


Figure 5: Male - Female Relationship visualized in a low dimensional vector space

- High dimensional **vector representation** of each word
- Used to **reconstruct linguistic context** of words
- **Capture semantic similarity** between words

Feature Extraction Process

- A conversation can be represented as a **set n word vectors** (n = number of unique words used in the conversation)

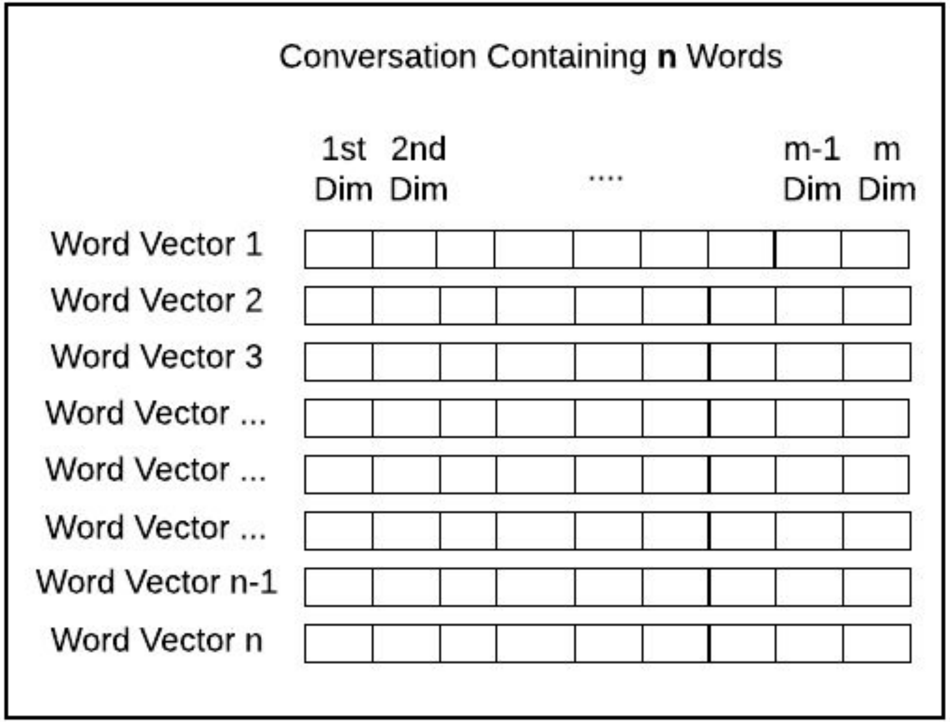


Figure 6: Set of n Word Vectors Representing a Conversation

Feature Extraction Process

- A conversation can be represented as a **set n word vectors** (n = number of unique words used in the conversation)

- Need to **extract features** from each conversation's word vectors in order to create **conversation feature vectors**

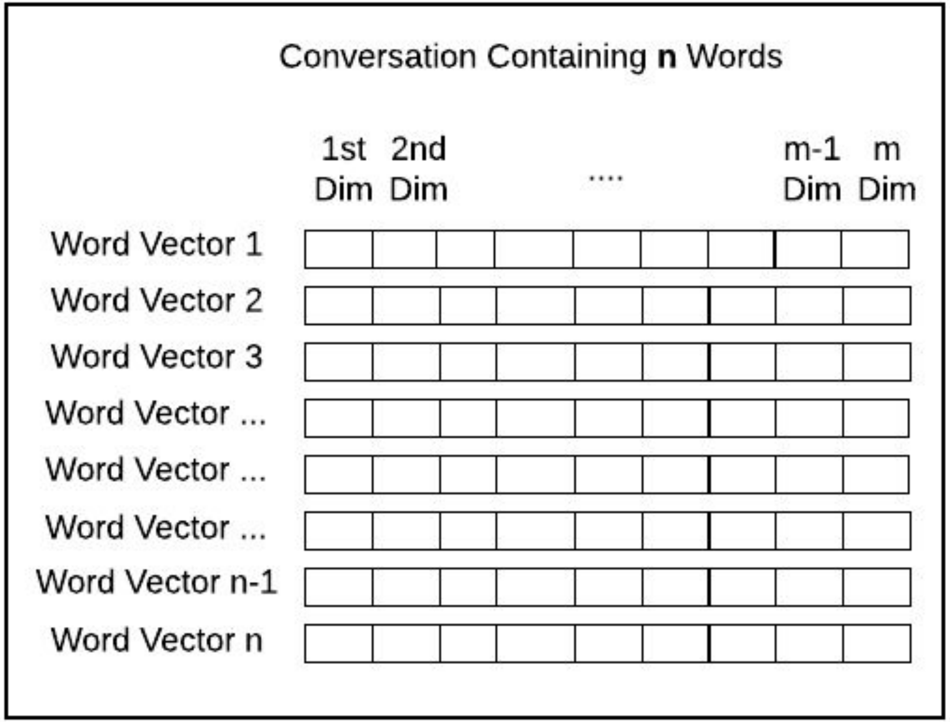


Figure 6: Set of n Word Vectors Representing a Conversation

Feature Extraction Process

- A conversation can be represented as a **set n word vectors**
(n = number of unique words used in the conversation)

- Need to **extract features** from each conversation's word vectors in order to create **conversation feature vectors**

- Used De Boom et. al (2016)'s **Coordinate-wise Word Vector Aggregation** technique as a Feature Extraction Process

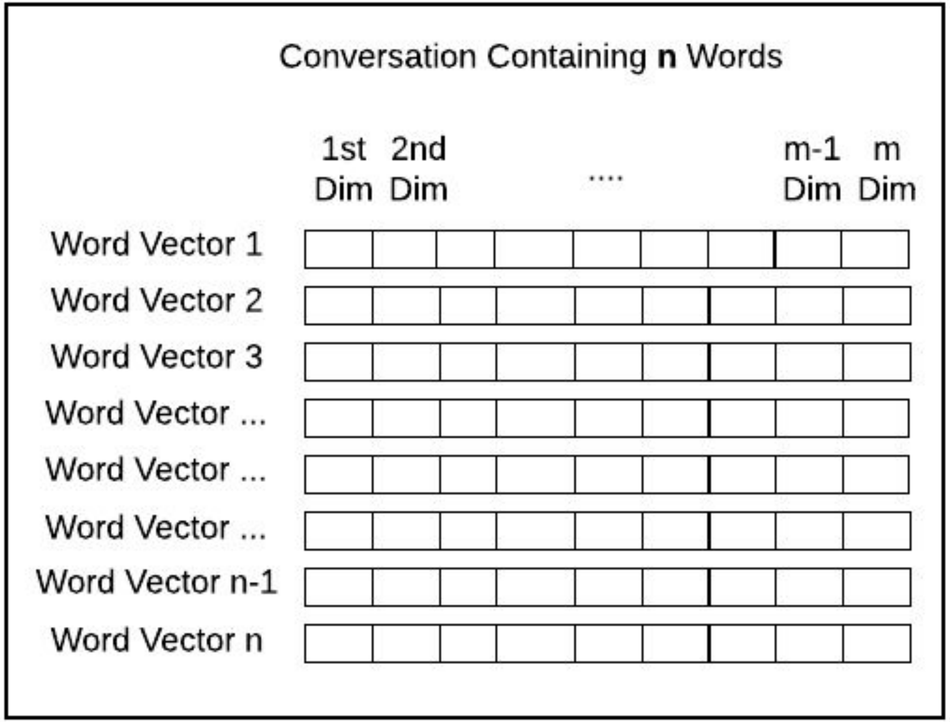


Figure 6: Set of n Word Vectors Representing a Conversation

Coordinate-wise Word Vector Aggregation

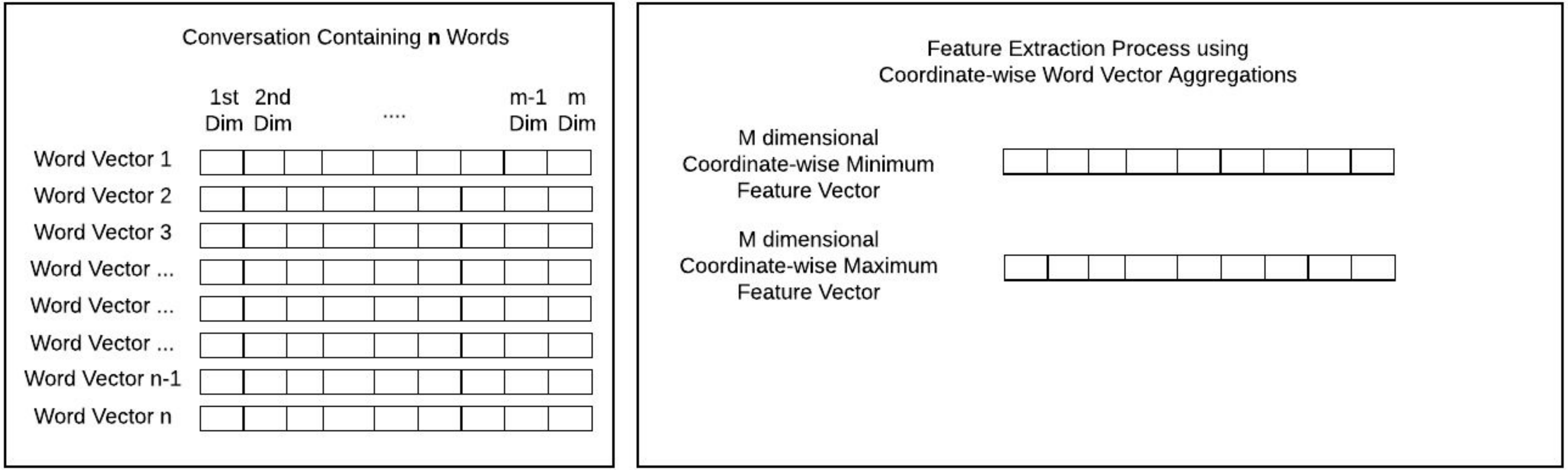


Figure 7: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

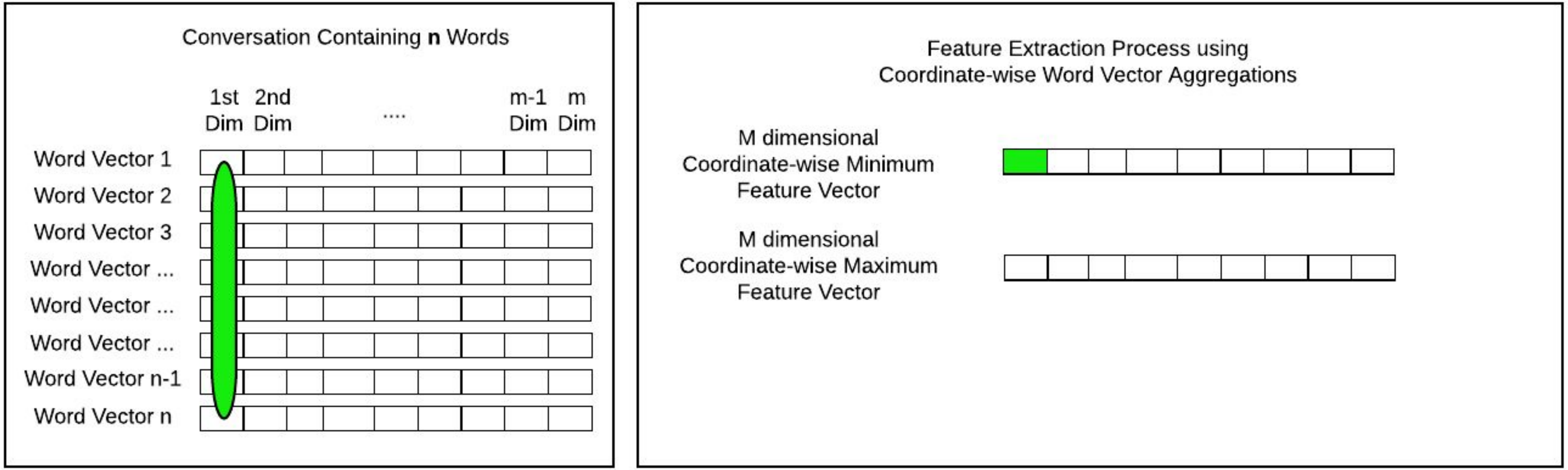


Figure 8: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

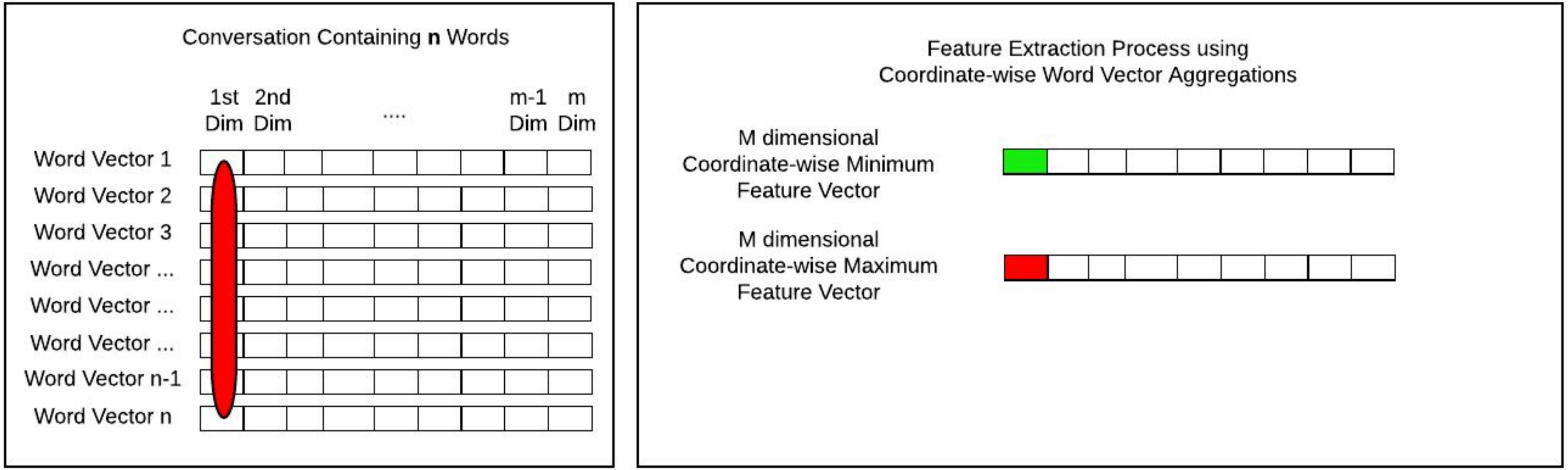


Figure 9: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

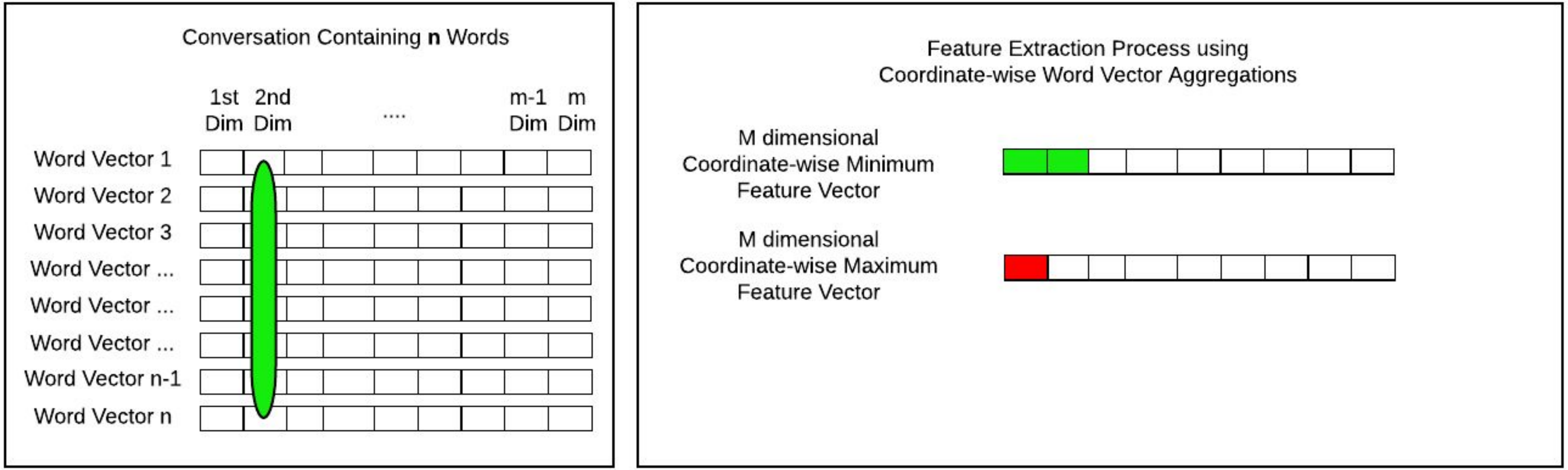


Figure 10: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

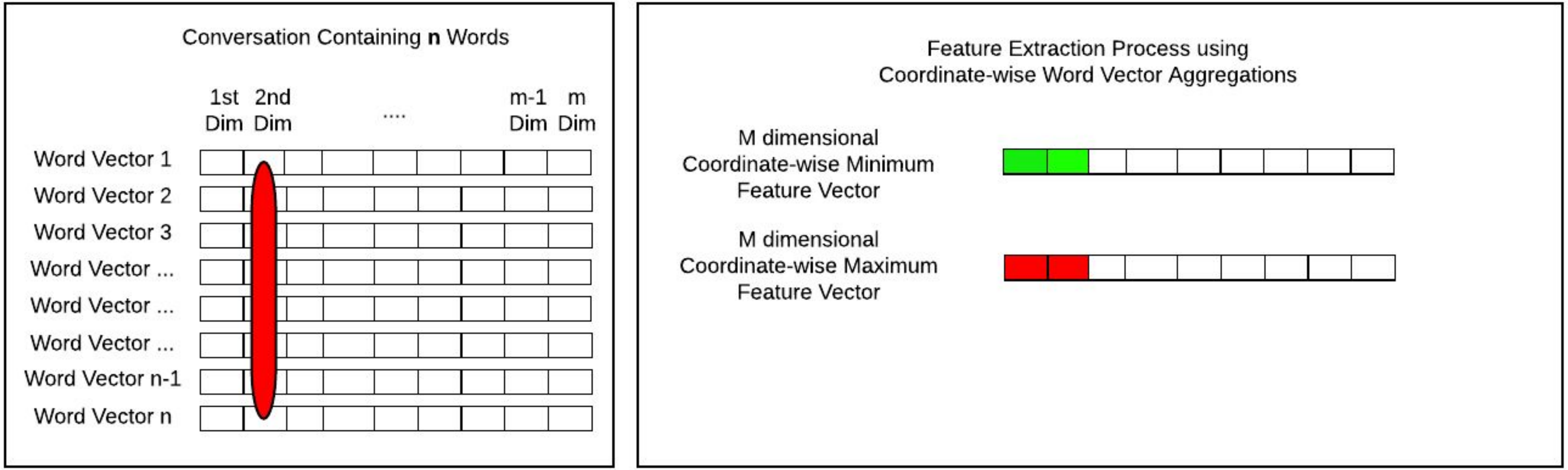


Figure 11: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

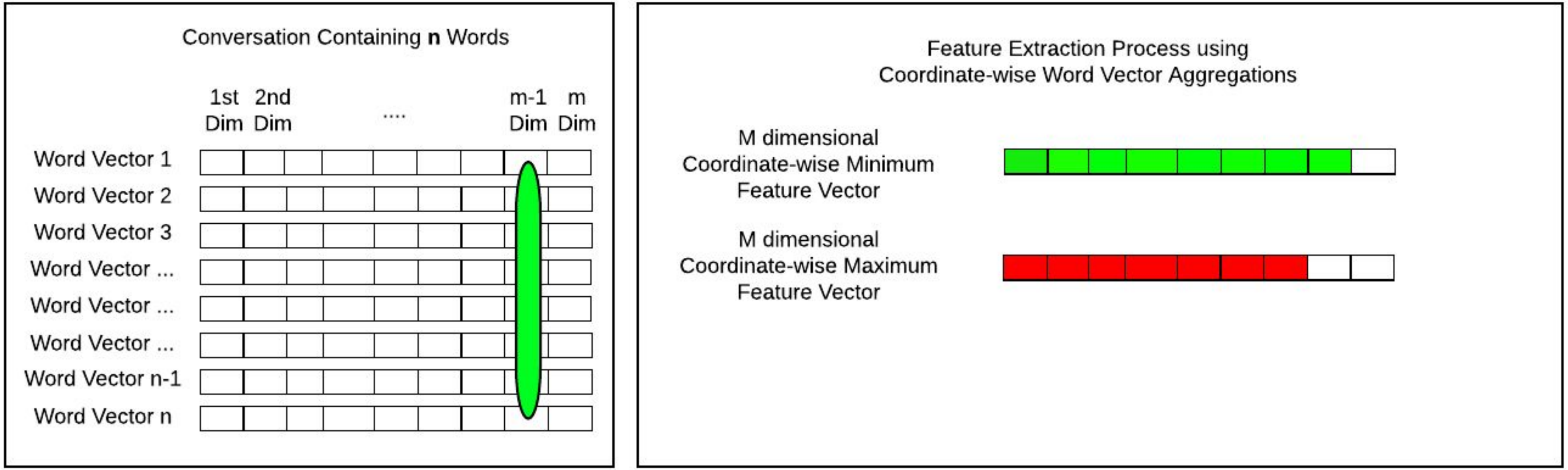


Figure 12: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

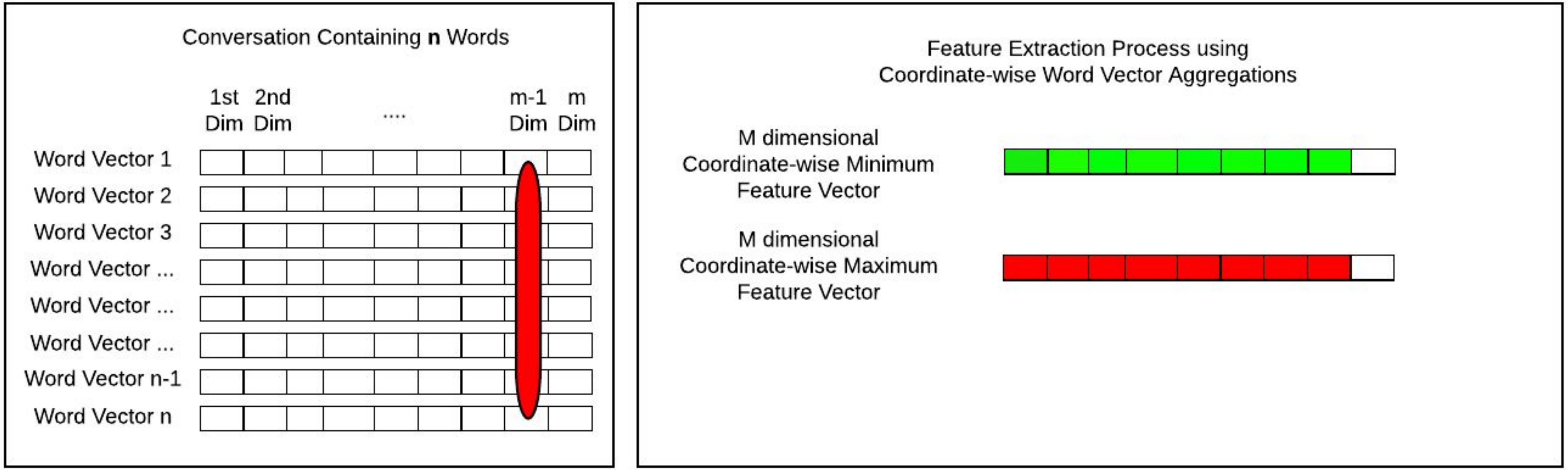


Figure 13: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

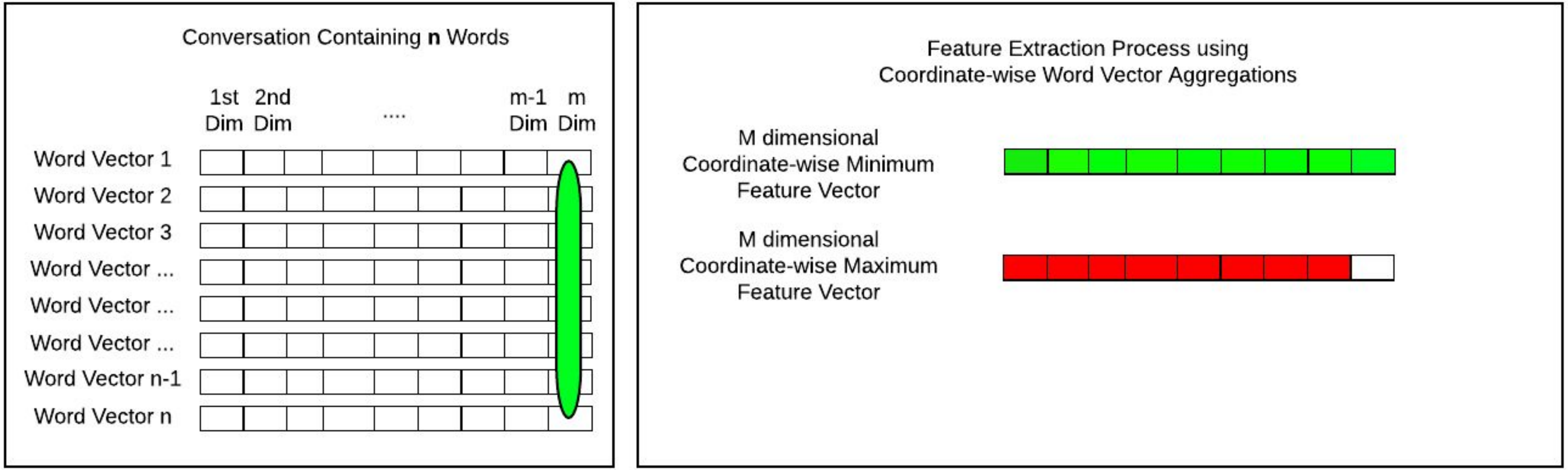


Figure 14: Feature Extraction Process

Coordinate-wise Word Vector Aggregation

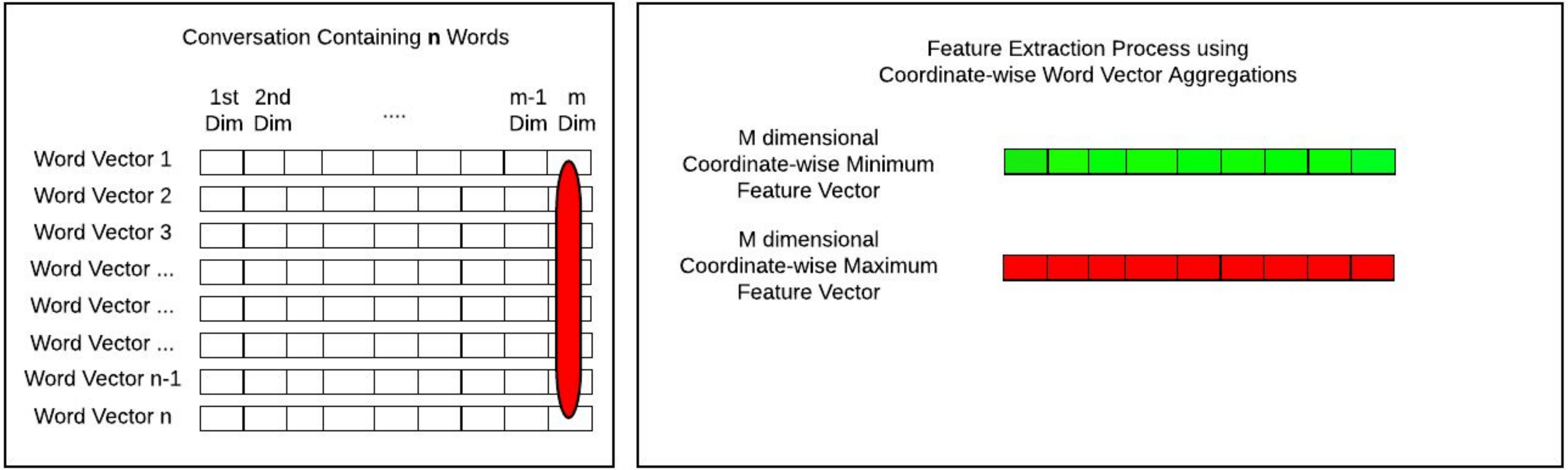


Figure 15: Feature Extraction Process

Conversation MIN-MAX Feature Vector

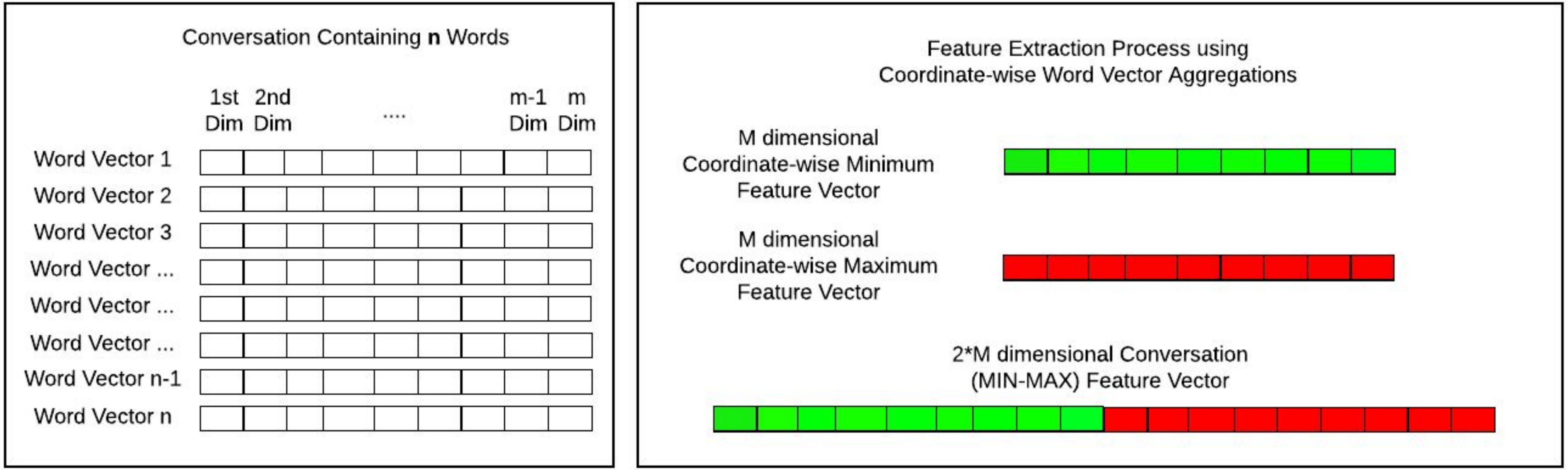


Figure 16: Feature Extraction Process

Sample Conversation Feature Vectors

1	dim1	dim2	dim3	dim4	dim5	dim6	dim7	dim8	dim9	dim10
2	0.326793	0.509899	0.319126	0.294877	0.830449	0.179753	0.158272	0.269415	0.158272	0.187153
3	0.233455	0.149392	0.18206	0.012931	0.281442	0.183503	0.236117	0.23604	0.710034	0.294323
4	0.455681	0.292429	0.268265	0.970831	0.393424	0.364359	0.261243	0.420074	0.255124	0.687304
5	0.435186	0.543371	0.421567	0.431468	0.415848	0.495029	0.476062	0.598359	0.427184	0.707913
6	0.395548	0.324849	0.500006	0.368869	0.741563	0.42329	0.25399	0.420074	0.427184	0.422575
7	0.240186	0.134993	0.123147	0.072401	0.239194	0.105477	0.186437	0.17187	0.308369	0.189388
8	0.137026	0.137568	0.285678	-0.05332	0.534698	0.364102	0.236117	0.089361	0.235917	0.502221
9	0.610423	0.238505	0.765492	0.39827	0.993194	0.112962	0.158272	0.816111	0.220825	0.138676
10	0.557104	0.265102	0.319126	0.212012	0.093506	0.286265	0.274407	0.376721	0.308369	0.435424
11	0.160055	0.413706	0.126338	0.377143	0.475433	0.312591	0.117461	0.562886	0.235337	0.262733
12	0.160055	0.155515	0.421567	0.136305	0.314459	0.210839	0.301609	0.271378	0.090368	0.074593
13	0.395548	0.41569	0.567015	0.487332	0.393424	0.341351	0.371596	0.816111	0.407664	0.327895
14	0.395548	0.335495	0.400459	0.424437	0.415848	0.437937	0.321841	0.547721	0.407664	0.488582
15	0.395548	0.112537	0.18206	0.163788	0.269591	0.247135	0.25399	0.420074	0.133222	0.422575
16	0.105116	0.494306	0.319126	0.316998	0.269591	0.352102	0.25399	0.420074	0.124105	0.422575
17	0.403466	0.335495	0.269617	0.321741	0.281442	0.43677	0.454666	0.816111	0.710034	0.437507
18	0.262715	0.18068	0.212783	-0.03116	0.203668	0.179753	0.113385	0.050921	0.144946	0.163538
19	0.503621	0.494306	0.213058	0.232814	0.393424	0.4216	0.25399	0.420074	0.308369	0.422575
20	0.112298	0.153483	0.18206	0.368869	0.237592	0.312591	0.236117	0.255418	-0.0176	0.294323
21	0.474968	0.317604	0.333141	0.459357	0.284466	0.505133	0.25399	0.420074	0.389327	0.422575

Figure 17: First 10 Dimensions of 20 Sample Conversation Feature Vectors

Sample Conversation Feature Vectors

1	dim1	dim2	dim3	dim4	dim5	dim6	dim7	dim8	dim9	dim10
2	0.326793	0.509899	0.319126	0.294877	0.830449	0.179753	0.158272	0.269415	0.158272	0.187153
3	0.233455	0.149392	0.18206	0.012931	0.281442	0.183503	0.236117	0.23604	0.710034	0.294323
4	0.455681	0.292429	0.268265	0.970831	0.393424	0.364359	0.261243	0.420074	0.255124	0.687304
5	0.435186	0.543371	0.421567	0.431468	0.415848	0.495029	0.476062	0.598359	0.427184	0.707913
6	0.395548	0.324849	0.500006	0.368869	0.741563	0.42329	0.25399	0.420074	0.427184	0.422575
7	0.240186	0.134993	0.123147	0.072401	0.239194	0.105477	0.186437	0.17187	0.308369	0.189388
8	0.137026	0.137568	0.285678	-0.05332	0.534698	0.364102	0.236117	0.089361	0.235917	0.502221
9	0.610423	0.238505	0.765492	0.39827	0.993194	0.112962	0.158272	0.816111	0.220825	0.138676
10	0.557104	0.265102	0.319126	0.212012	0.093506	0.286265	0.274407	0.376721	0.308369	0.435424
11	0.160055	0.413706	0.126338	0.377143	0.475433	0.312591	0.117461	0.562886	0.235337	0.262733
12	0.160055	0.155515	0.421567	0.136305	0.314459	0.210839	0.301609	0.271378	0.090368	0.074593
13	0.395548	0.41569	0.567015	0.487332	0.393424	0.341351	0.371596	0.816111	0.407664	0.327895
14	0.395548	0.335495	0.400459	0.424437	0.415848	0.437937	0.321841	0.547721	0.407664	0.488582
15	0.395548	0.112537	0.18206	0.163788	0.269591	0.247135	0.25399	0.420074	0.133222	0.422575
16	0.105116	0.494306	0.319126	0.316998	0.269591	0.352102	0.25399	0.420074	0.124105	0.422575
17	0.403466	0.335495	0.269617	0.321741	0.281442	0.43677	0.454666	0.816111	0.710034	0.437507
18	0.262715	0.18068	0.212783	-0.03116	0.203668	0.179753	0.113385	0.050921	0.144946	0.163538
19	0.503621	0.494306	0.213058	0.232814	0.393424	0.4216	0.25399	0.420074	0.308369	0.422575
20	0.112298	0.153483	0.18206	0.368869	0.237592	0.312591	0.236117	0.255418	-0.0176	0.294323
21	0.474968	0.317604	0.333141	0.459357	0.284466	0.505133	0.25399	0.420074	0.389327	0.422575

Figure 17: First 10 Dimensions of 20 Sample Conversation Feature Vectors

IN REALITY

My Word Vectors are **400 dimensional**

Sample Conversation Feature Vectors

1	dim1	dim2	dim3	dim4	dim5	dim6	dim7	dim8	dim9	dim10
2	0.326793	0.509899	0.319126	0.294877	0.830449	0.179753	0.158272	0.269415	0.158272	0.187153
3	0.233455	0.149392	0.18206	0.012931	0.281442	0.183503	0.236117	0.23604	0.710034	0.294323
4	0.455681	0.292429	0.268265	0.970831	0.393424	0.364359	0.261243	0.420074	0.255124	0.687304
5	0.435186	0.543371	0.421567	0.431468	0.415848	0.495029	0.476062	0.598359	0.427184	0.707913
6	0.395548	0.324849	0.500006	0.368869	0.741563	0.42329	0.25399	0.420074	0.427184	0.422575
7	0.240186	0.134993	0.123147	0.072401	0.239194	0.105477	0.186437	0.17187	0.308369	0.189388
8	0.137026	0.137568	0.285678	-0.05332	0.534698	0.364102	0.236117	0.089361	0.235917	0.502221
9	0.610423	0.238505	0.765492	0.39827	0.993194	0.112962	0.158272	0.816111	0.220825	0.138676
10	0.557104	0.265102	0.319126	0.212012	0.093506	0.286265	0.274407	0.376721	0.308369	0.435424
11	0.160055	0.413706	0.126338	0.377143	0.475433	0.312591	0.117461	0.562886	0.235337	0.262733
12	0.160055	0.155515	0.421567	0.136305	0.314459	0.210839	0.301609	0.271378	0.090368	0.074593
13	0.395548	0.41569	0.567015	0.487332	0.393424	0.341351	0.371596	0.816111	0.407664	0.327895
14	0.395548	0.335495	0.400459	0.424437	0.415848	0.437937	0.321841	0.547721	0.407664	0.488582
15	0.395548	0.112537	0.18206	0.163788	0.269591	0.247135	0.25399	0.420074	0.133222	0.422575
16	0.105116	0.494306	0.319126	0.316998	0.269591	0.352102	0.25399	0.420074	0.124105	0.422575
17	0.403466	0.335495	0.269617	0.321741	0.281442	0.43677	0.454666	0.816111	0.710034	0.437507
18	0.262715	0.18068	0.212783	-0.03116	0.203668	0.179753	0.113385	0.050921	0.144946	0.163538
19	0.503621	0.494306	0.213058	0.232814	0.393424	0.4216	0.25399	0.420074	0.308369	0.422575
20	0.112298	0.153483	0.18206	0.368869	0.237592	0.312591	0.236117	0.255418	-0.0176	0.294323
21	0.474968	0.317604	0.333141	0.459357	0.284466	0.505133	0.25399	0.420074	0.389327	0.422575

Figure 17: First 10 Dimensions of 20 Sample Conversation Feature Vectors

IN REALITY

My Word Vectors are **400 dimensional**

MIN-MAX Feature Vectors are **800 dimensional**

Sample Conversation Feature Vectors

1	dim1	dim2	dim3	dim4	dim5	dim6	dim7	dim8	dim9	dim10
2	0.326793	0.509899	0.319126	0.294877	0.830449	0.179753	0.158272	0.269415	0.158272	0.187153
3	0.233455	0.149392	0.18206	0.012931	0.281442	0.183503	0.236117	0.23604	0.710034	0.294323
4	0.455681	0.292429	0.268265	0.970831	0.393424	0.364359	0.261243	0.420074	0.255124	0.687304
5	0.435186	0.543371	0.421567	0.431468	0.415848	0.495029	0.476062	0.598359	0.427184	0.707913
6	0.395548	0.324849	0.500006	0.368869	0.741563	0.42329	0.25399	0.420074	0.427184	0.422575
7	0.240186	0.134993	0.123147	0.072401	0.239194	0.105477	0.186437	0.17187	0.308369	0.189388
8	0.137026	0.137568	0.285678	-0.05332	0.534698	0.364102	0.236117	0.089361	0.235917	0.502221
9	0.610423	0.238505	0.765492	0.39827	0.993194	0.112962	0.158272	0.816111	0.220825	0.138676
10	0.557104	0.265102	0.319126	0.212012	0.093506	0.286265	0.274407	0.376721	0.308369	0.435424
11	0.160055	0.413706	0.126338	0.377143	0.475433	0.312591	0.117461	0.562886	0.235337	0.262733
12	0.160055	0.155515	0.421567	0.136305	0.314459	0.210839	0.301609	0.271378	0.090368	0.074593
13	0.395548	0.41569	0.567015	0.487332	0.393424	0.341351	0.371596	0.816111	0.407664	0.327895
14	0.395548	0.335495	0.400459	0.424437	0.415848	0.437937	0.321841	0.547721	0.407664	0.488582
15	0.395548	0.112537	0.18206	0.163788	0.269591	0.247135	0.25399	0.420074	0.133222	0.422575
16	0.105116	0.494306	0.319126	0.316998	0.269591	0.352102	0.25399	0.420074	0.124105	0.422575
17	0.403466	0.335495	0.269617	0.321741	0.281442	0.43677	0.454666	0.816111	0.710034	0.437507
18	0.262715	0.18068	0.212783	-0.03116	0.203668	0.179753	0.113385	0.050921	0.144946	0.163538
19	0.503621	0.494306	0.213058	0.232814	0.393424	0.4216	0.25399	0.420074	0.308369	0.422575
20	0.112298	0.153483	0.18206	0.368869	0.237592	0.312591	0.236117	0.255418	-0.0176	0.294323
21	0.474968	0.317604	0.333141	0.459357	0.284466	0.505133	0.25399	0.420074	0.389327	0.422575

Figure 17: First 10 Dimensions of 20 Sample Conversation Feature Vectors

IN REALITY

My Word Vectors are **400 dimensional**

MIN-MAX Feature Vectors are **800 dimensional**

For 100.000 conversations ...
DATA = **100.000 x 800 matrix**

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

- Second stage classification: Train an **AdaBoost Binary Classifier** on all observations labelled as predatory **to filter out the LDA Classifier's misclassifications**

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

- Second stage classification: Train an **AdaBoost Binary Classifier** on all observations labelled as predatory **to filter out the LDA Classifier's misclassifications**

- Obtain 3 groups of conversations with different levels of maliciousness:

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

- Second stage classification: Train an **AdaBoost Binary Classifier** on all observations labelled as predatory **to filter out the LDA Classifier's misclassifications**

- Obtain 3 groups of conversations with different levels of maliciousness:
  - Group A = conversations most likely do not contain predatory behaviour

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

- Second stage classification: Train an **AdaBoost Binary Classifier** on all observations labelled as predatory **to filter out the LDA Classifier's misclassifications**

- Obtain 3 groups of conversations with different levels of maliciousness:
  - Group A = conversations most likely do not contain predatory behaviour
  - Group B = conversations possibly could contain predatory behaviour

Two Stage Classification System

- First stage classification: Train a **Linear Discriminant Analysis Binary Classifier** on conversation feature vectors

- Second stage classification: Train an **AdaBoost Binary Classifier** on all observations labelled as predatory **to filter out the LDA Classifier's misclassifications**

- Obtain 3 groups of conversations with different levels of maliciousness:
  - Group A = conversations most likely do not contain predatory behaviour
  - Group B = conversations possibly could contain predatory behaviour
  - Group C = conversations most likely contain predatory behaviour

Algorithm Highlights

1. **Captures contextual details** by putting an emphasis on insight that lies within the conversation

Algorithm Highlights

1. **Captures contextual details** by putting an emphasis on insight that lies within the conversation

2. Uses a **domain specific feature extraction technique** that **extracts the essential details** from each conversation

Algorithm Highlights

1. **Captures contextual details** by putting an emphasis on insight that lies within the conversation

2. Uses a **domain specific feature extraction technique** that **extracts the essential details** from each conversation

3. Creates a highly flexible and customizable **two stage classification system** for detecting and classifying conversations containing sexual predatory behaviour

Outline

1.) Problem Background
2.) Research Questions and Objectives
3.) Introduction to Data Being Used
4.) Algorithm Proposed
5.) Results & Model Selection Explained
6.) Future Works and Conclusion

k-Fold Cross Validation for Results

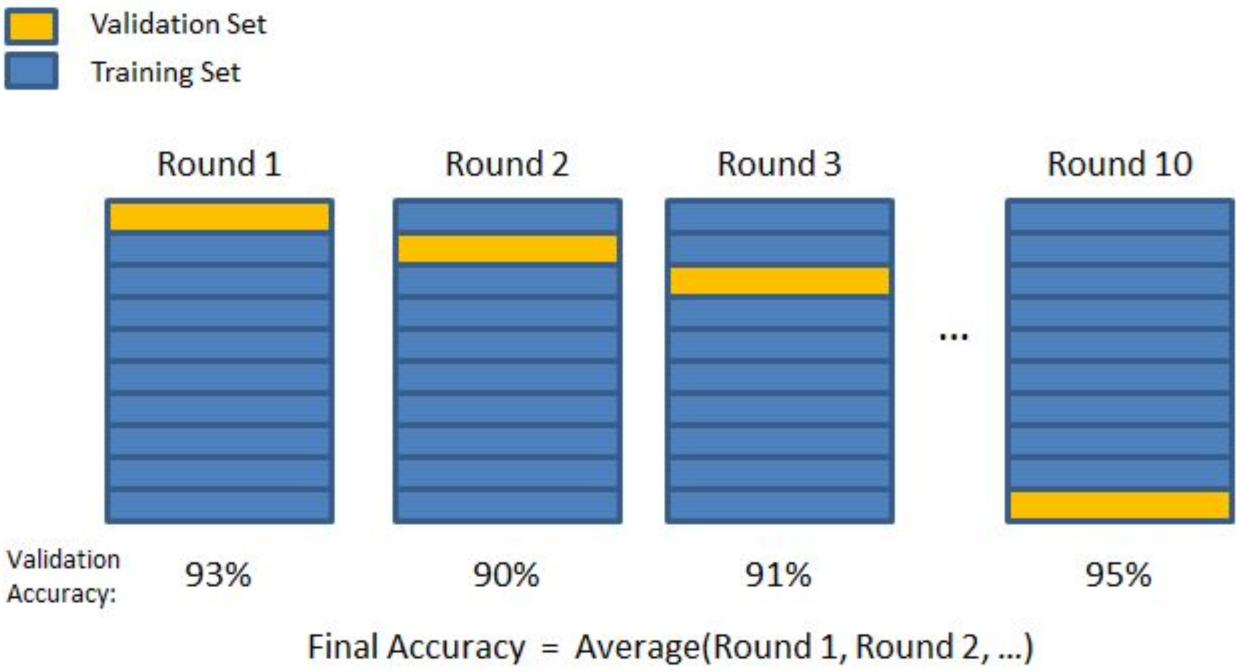


Figure 18: k-fold Cross Validation Visualization

- k = 10 folds
- Dataset contained 155128 conversations
- Took sample of 100000 conversations
- 90000 training set
- 10000 validation set

First Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

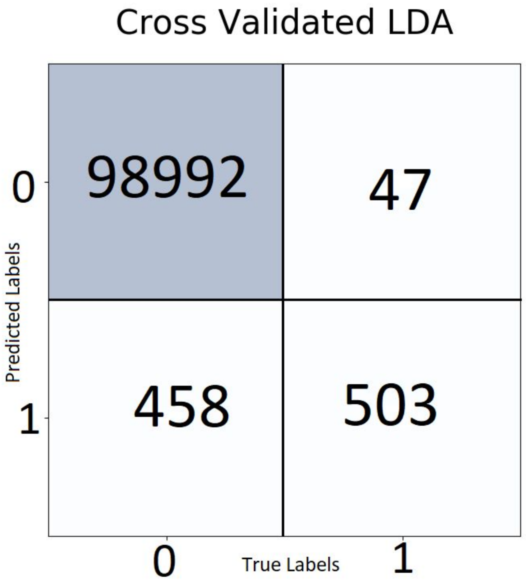
First Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

0 = **non-predatory** behavior

1 = **predatory** behavior



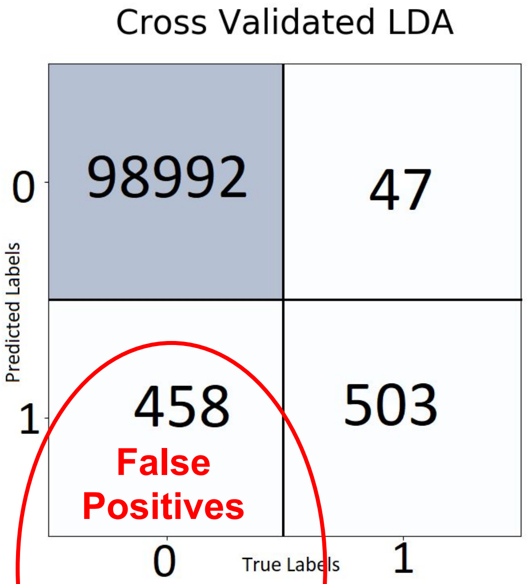
First Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

0 = **non-predatory** behavior

1 = **predatory** behavior



Non-predatory conversations **labelled as Predatory**

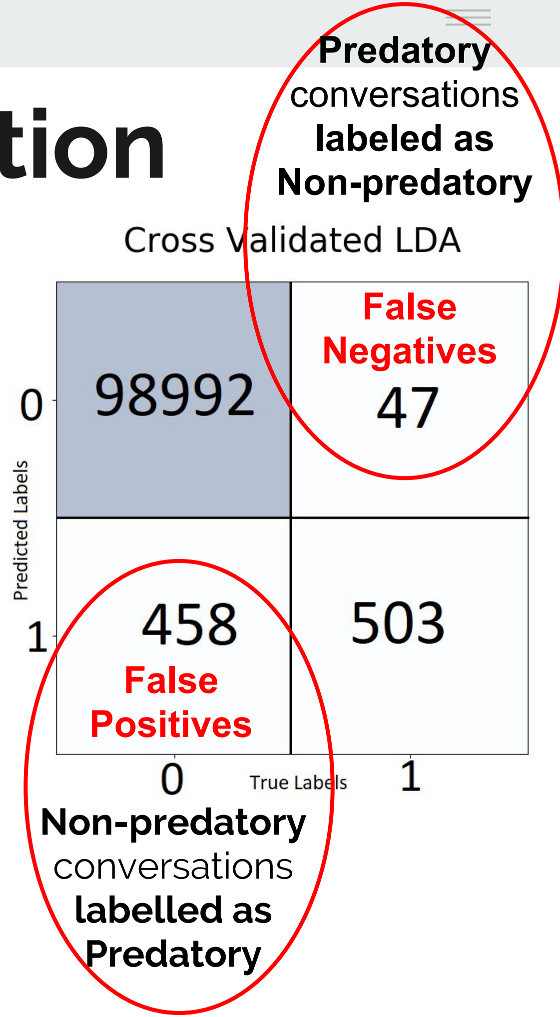
First Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

0 = **non-predatory** behavior

1 = **predatory** behavior



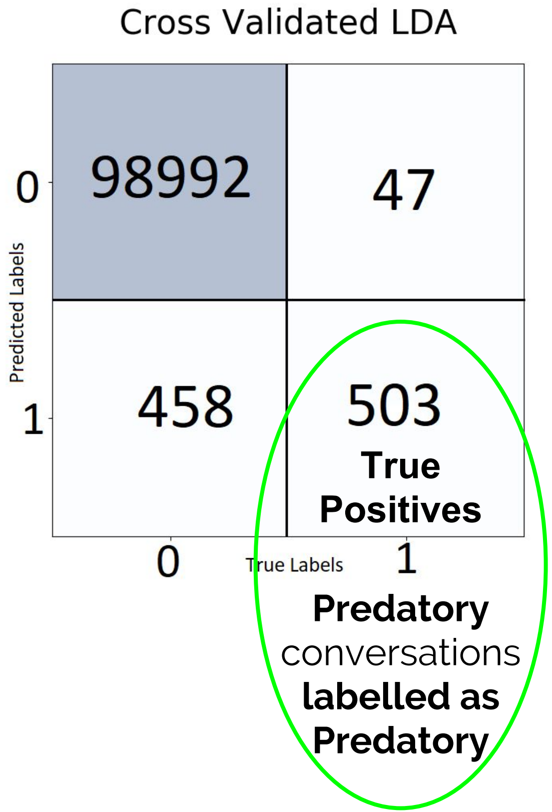
First Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

0 = **non-predatory** behavior

1 = **predatory** behavior



First Stage Classification

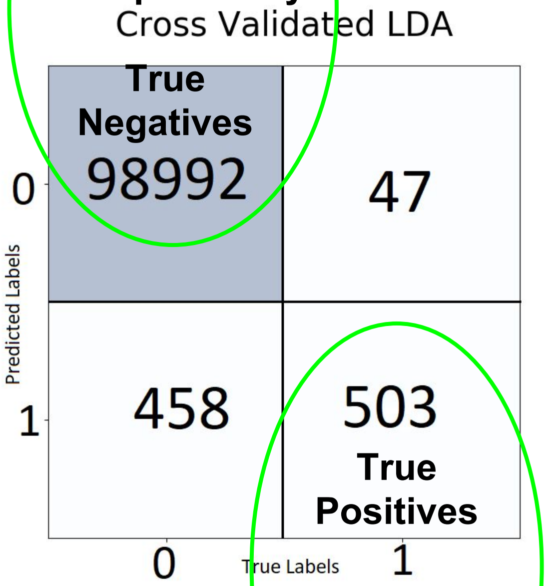
Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

0 = **non-predatory** behavior

1 = **predatory** behavior

Non-predatory conversations **labeled as Non-predatory**



Predatory conversations **labelled as Predatory**

LDA - Recall

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

Recall = When it's actually predatory, how often does it predict predatory?

Recall = TP / (TP + FN)

= 503 / (503 + 47)

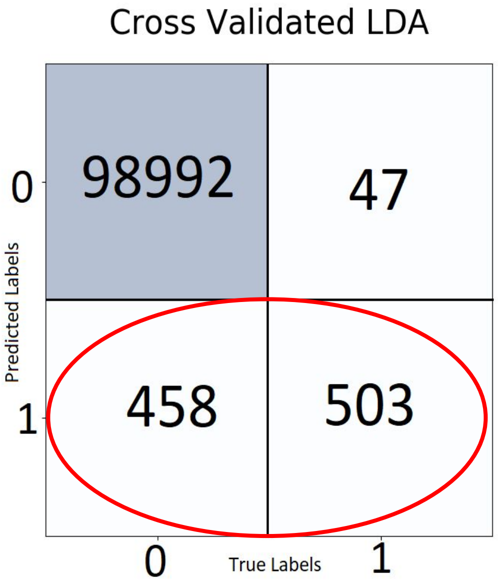
= 0.9145

LDA - Precision

Classification Model	Average Precision	Average Recall	F1 Score
LDA	0.5234	0.9145	0.6658
SVM	0.7686	0.6582	0.7091
Random Forest	0.8241	0.2982	0.4379
LASSO	0.6739	0.6492	0.6613
Gr. Boosting Machine	0.8646	0.3018	0.4474

Table 1: Average Recall and Precision, and Overall F1-Score from the top 5 First Stage Classification Models.

Precision = When it predicts Predatory, how often is it correct?



Precision = TP / (TP + FP)

= 503 / (503 + 458)

= 0.5234

What is LDA ? Why does it work well?

- LDA = Linear Discriminant Analysis

What is LDA ? Why does it work well?

- **LDA = Linear Discriminant Analysis**

- **Dimensionality reduction method** that can be used for **classification tasks**

What is LDA ? Why does it work well?

- **LDA = Linear Discriminant Analysis**
- **Dimensionality reduction method** that can be used for **classification tasks**
- Standard implementation assumes a **Gaussian distribution** of the input variables

What is LDA ? Why does it work well?

- **LDA = Linear Discriminant Analysis**

- **Dimensionality reduction method** that can be used for **classification tasks**

- Standard implementation assumes a **Gaussian distribution** of the input variables

- Main concept behind LDA is **searching for a linear combination of variables** that best separates two (or more) classes

What is LDA ? Why does it work well?

- **LDA = Linear Discriminant Analysis**

- **Dimensionality reduction method** that can be used for **classification tasks**

- Standard implementation assumes a **Gaussian distribution** of the input variables

- Main concept behind LDA is **searching for a linear combination of variables** that best separates two (or more) classes

Discriminant Function:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Where
x is an observation (vector of input variables)
$\hat{\mu}_k$ is the estimated mean of the group k
$\hat{\sigma}^2$ is the estimated variance
$\hat{\pi}_k$ is the prior class membership probability of a group k

What is LDA ? Why does it work well?

- **LDA = Linear Discriminant Analysis**
- **Dimensionality reduction method** that can be used for **classification tasks**
- Standard implementation assumes a **Gaussian distribution** of the input variables
- Main concept behind LDA is **searching for a linear combination of variables** that best separates two (or more) classes

Discriminant Function:

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

Where
x is an observation (vector of input variables)
$\hat{\mu}_k$ is the estimated mean of the group k
$\hat{\sigma}^2$ is the estimated variance
$\hat{\pi}_k$ is the prior class membership probability of a group k $\Longrightarrow$ In my LDA: $\hat{\pi}_0 = 0.99450,$ $\hat{\pi}_1 = 0.00550.$

The observation X = x gets assigned to the class k for which $\hat{\delta}_k(x)$ is largest.

Second Stage Classification

Classification Model	Average Precision	Average Recall	F1 Score
SVM	0.6928	0.8250	0.7532
Naive Bayes	0.5859	0.9284	0.7185
LASSO	0.7433	0.7714	0.7571
AdaBoost	0.7767	0.8091	0.7926
k-NN	0.6273	0.8966	0.7381

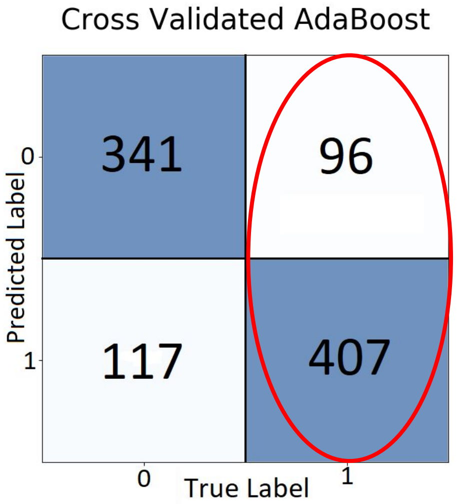
Table 2: Second stage classification process, with cross validated results from the top 5 models, and their average precision, recall measurements, alongside F1-scores.

AdaBoost - Recall

Classification Model	Average Precision	Average Recall	F1 Score
SVM	0.6928	0.8250	0.7532
Naive Bayes	0.5859	0.9284	0.7185
LASSO	0.7433	0.7714	0.7571
AdaBoost	0.7767	0.8091	0.7926
k-NN	0.6273	0.8966	0.7381

Table 2: Second stage classification process, with cross validated results from the top 5 models, and their average precision, recall measurements, alongside F1-scores.

Recall = When it's actually predatory, how often does it predict predatory?



Recall = TP / (TP + FN)

= 407 / (407 + 96)

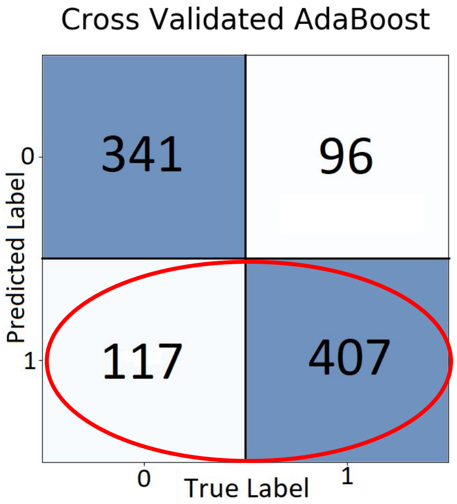
= 0.8091

AdaBoost - Precision

Classification Model	Average Precision	Average Recall	F1 Score
SVM	0.6928	0.8250	0.7532
Naive Bayes	0.5859	0.9284	0.7185
LASSO	0.7433	0.7714	0.7571
AdaBoost	0.7767	0.8091	0.7926
k-NN	0.6273	0.8966	0.7381

Table 2: Second stage classification process, with cross validated results from the top 5 models, and their average precision, recall measurements, alongside F1-scores.

Precision = When it predicts Predatory, how often is it correct?



Precision = TP / (TP + FP)

= 407 / (407 + 117)

= 0.7767

What is AdaBoost ?

- AdaBoost (Adaptive Boosting) is an iterative tree based method

What is AdaBoost ?

- AdaBoost (Adaptive Boosting) is an iterative tree based method

- Simple variation on **Bagging algorithm**

What is AdaBoost ?

- AdaBoost (Adaptive Boosting) is an iterative tree based method
- Simple variation on **Bagging algorithm**
- Improves the learning process where the system is not performing well

What is AdaBoost ?

- AdaBoost (Adaptive Boosting) is an iterative tree based method
- Simple variation on **Bagging algorithm**
- Improves the learning process where the system is not performing well
- Main concept: **Iteratively learns from misclassifications** of previously fitted models

What is AdaBoost ?

AdaBoost Pseudo-Algorithm:

N observations, M number of trees (bags)

Initialize observation weights $w_i = 1/N$

What is AdaBoost ?

AdaBoost Pseudo-Algorithm:

N observations, M number of trees (bags)

Initialize observation weights $w_i = 1/N$

Randomly split data into training and test set

What is AdaBoost ?

AdaBoost Pseudo-Algorithm:

N observations, M number of trees (bags)

Initialize observation weights $w_i = 1/N$
Randomly split data into training and test set
For m = 1 to M:

Fit new model using m trees on training data **(weak classifier)**
Combine and test new model with previously fitted models
Compute error and accuracy for the **combined models**
Update observation weights (correct classifications get less weightage, **misclassifications' weights increase**)
Create new train/test sets randomly* from original data
(*misclassifications have a higher chance of being in the new training set)

What is AdaBoost ?

AdaBoost Pseudo-Algorithm:

N observations, M number of trees (bags)

Initialize observation weights $w_i = 1/N$
Randomly split data into training and test set
For m = 1 to M:

Fit new model using m trees on training data **(weak classifier)**
Combine and test new model with previously fitted models
Compute error and accuracy for the **combined models**
Update observation weights (correct classifications get less weightage, **misclassifications' weights increase**)
Create new train/test sets randomly* from original data
(*misclassifications have a higher chance of being in the new training set)

Combine all fitted models based on weight values to create AdaBoost Classifier

Visualization of AdaBoost

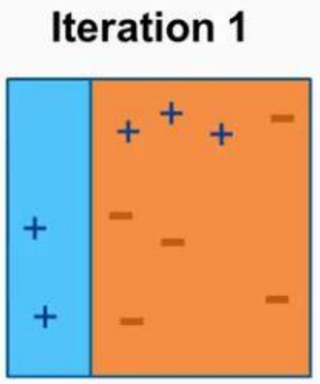


Figure 19: AdaBoost - First Iteration

Visualization of AdaBoost

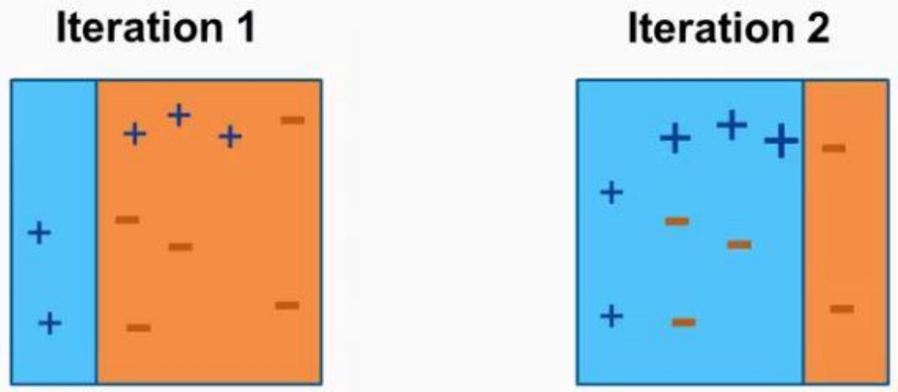


Figure 20: AdaBoost - Second Iteration

Visualization of AdaBoost

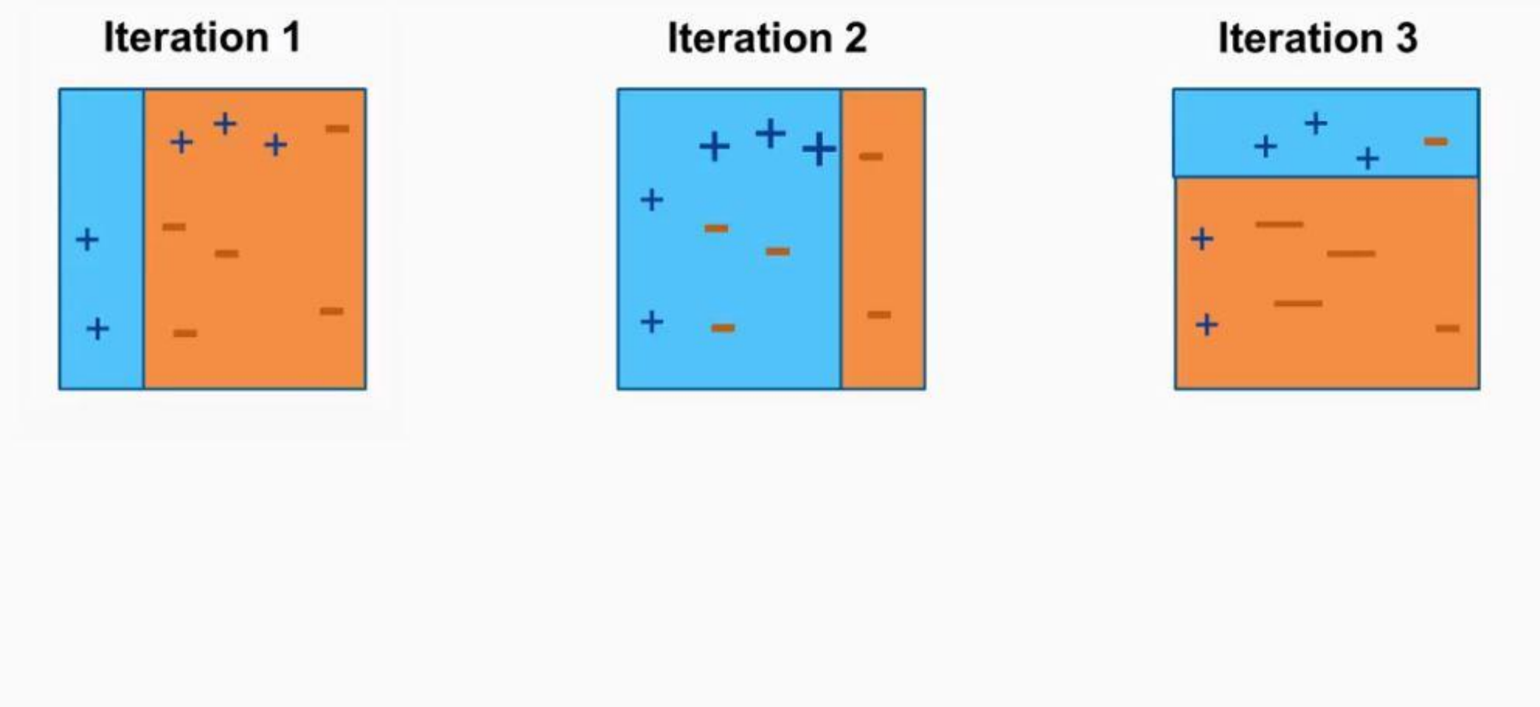


Figure 21: AdaBoost - Third Iteration

Visualization of AdaBoost

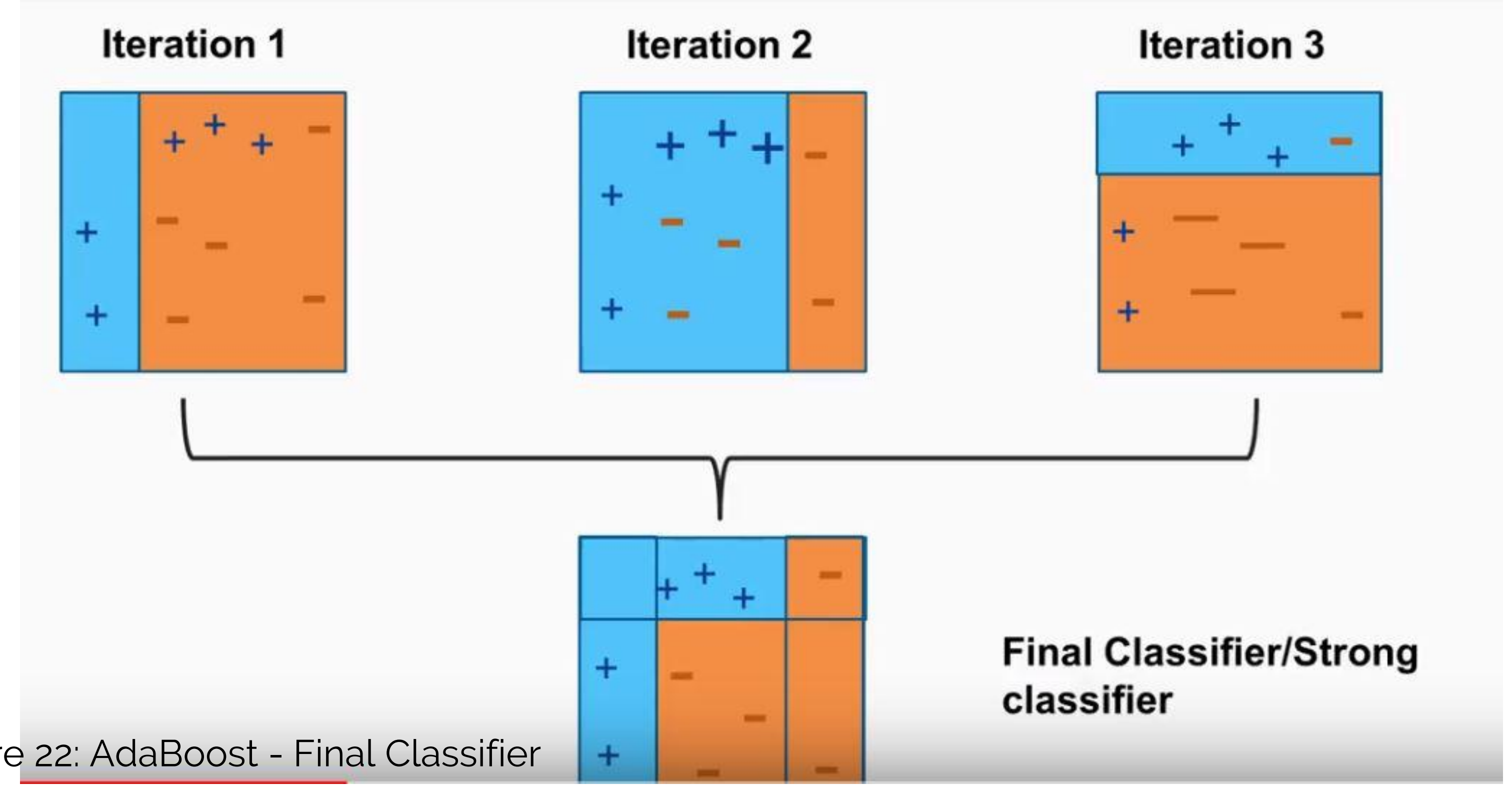


Figure 22: AdaBoost - Final Classifier

Classification System: LDA + AdaBoost

System of Classifiers	Precision	Recall	F1 Score
LDA $\rightarrow$ SVM	0.6928	0.7545	0.7224
LDA $\rightarrow$ Naive Bayes	0.5859	0.8491	0.6934
LDA $\rightarrow$ AdaBoost	0.7767	0.74	0.7579
LDA $\rightarrow$ LASSO	0.7433	0.7055	0.7239
LDA $\rightarrow$ k-NN	0.6273	0.82	0.7108

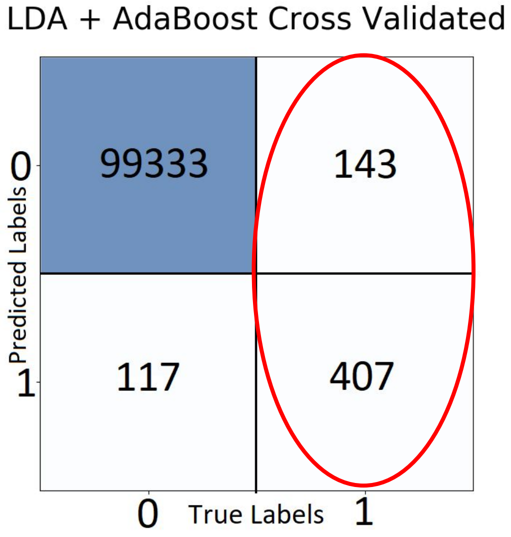
Table 3: Recall, precision and F1 scores from First Stage classifier combined with each possible second stage classifier.

LDA + AdaBoost Overall Recall

System of Classifiers	Precision	Recall	F1 Score
LDA → SVM	0.6928	0.7545	0.7224
LDA → Naive Bayes	0.5859	0.8491	0.6934
LDA → AdaBoost	0.7767	0.74	0.7579
LDA → LASSO	0.7433	0.7055	0.7239
LDA → k-NN	0.6273	0.82	0.7108

Table 3: Recall, precision and F1 scores from First Stage classifier combined with each possible second stage classifier.

Recall = When it's actually predatory, how often does it predict predatory?



Recall = TP / (TP + FN)

= 407 / (407 + 143)

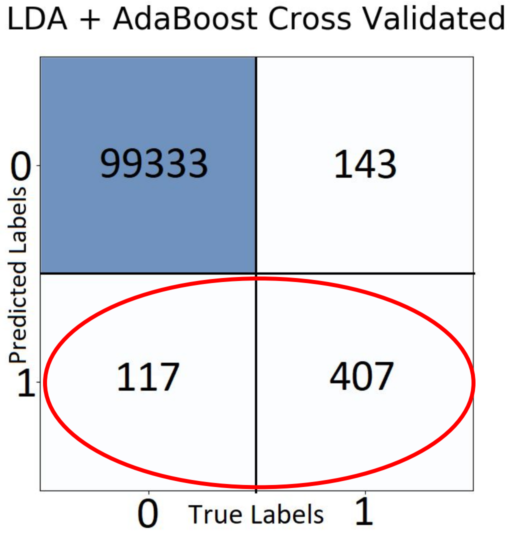
= 0.74

LDA + AdaBoost Overall Precision

System of Classifiers	Precision	Recall	F1 Score
LDA → SVM	0.6928	0.7545	0.7224
LDA → Naive Bayes	0.5859	0.8491	0.6934
LDA → AdaBoost	0.7767	0.74	0.7579
LDA → LASSO	0.7433	0.7055	0.7239
LDA → k-NN	0.6273	0.82	0.7108

Table 3: Recall, precision and F1 scores from First Stage classifier combined with each possible second stage classifier.

Precision = When it predicts Predatory, how often is it correct?



Precision = TP / (TP + FP)

= 407 / (407 + 117)

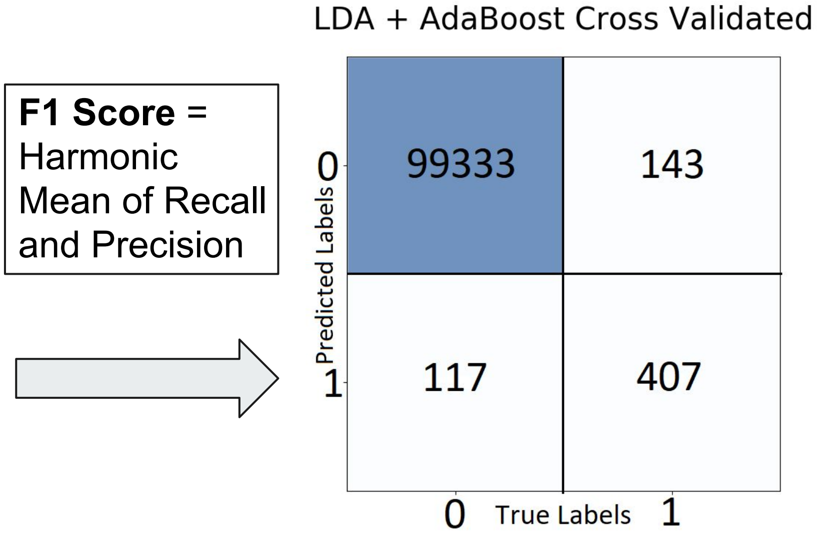
= 0.7767

LDA + AdaBoost Overall F1 Score

System of Classifiers	Precision	Recall	F1 Score
LDA → SVM	0.6928	0.7545	0.7224
LDA → Naive Bayes	0.5859	0.8491	0.6934
LDA → AdaBoost	0.7767	0.74	0.7579
LDA → LASSO	0.7433	0.7055	0.7239
LDA → k-NN	0.6273	0.82	0.7108

Table 3: Recall, precision and F1 scores from First Stage classifier combined with each possible second stage classifier.

F1 Score = Harmonic Mean of Recall and Precision



F1 Score = 2 * (Recall * Precision) / (Recall + Precision)

= 0.7579

Research Contribution

1. Research is focused on analyzing the entire conversation and **putting an emphasis on insight** that lies within the **contextual details** of a conversation.

Research Contribution

1. Research is focused on analyzing the entire conversation and **putting an emphasis on insight** that lies within the **contextual details** of a conversation.

2. The **experimentation process of two stage classification** system yielded performance results from 8 different classification models.

Research Significance

1. **Helps online communities** to enhance their member's safety by detecting malicious conversations of sexual nature.

Research Significance

1. **Helps online communities** to enhance their member's safety by detecting malicious conversations of sexual nature.

2. The algorithm designed **furthers research** conducted in the area of sexual predator detection.

Research Significance

1. **Helps online communities** to enhance their member's safety by detecting malicious conversations of sexual nature.

2. The algorithm designed **furthers research** conducted in the area of sexual predator detection.

3. Two stage classification system is a **highly flexible** method, therefore future research can be focused on **customizing this approach** to other types of dangerous behavior detection.

Outline

1.) Problem Background
2.) Research Questions and Objectives
3.) Introduction to Data Being Used
4.) Algorithm Proposed
5.) Results & Model Selection Explained
6.) **Future Works and Conclusion**

Future Works

1. Use FastText, Facebook's **efficient learning of word representations** model, **to improve the quality of the word vectors** (Joulin et al., 2016).

Future Works

1. Use FastText, Facebook's **efficient learning of word representations** model, **to improve the quality of the word vectors** (Joulin et al., 2016).

2. **Apply the entire version** of De Boom et al. (2016)'s **representation learning algorithm** combined with weighted word embedding aggregation.

Future Works

1. Use FastText, Facebook's **efficient learning of word representations** model, **to improve the quality of the word vectors** (Joulin et al., 2016).

2. **Apply the entire version** of De Boom et al. (2016)'s **representation learning algorithm** combined with weighted word embedding aggregation.

3. A **deep learning approach for the classification system** could be considered, by favouring prediction accuracy over model interpretability.

Conclusion

- Project was the perfect combination of **computational linguistics** and **statistical machine learning**

Conclusion

- Project was the perfect combination of **computational linguistics** and **statistical machine learning**

- The **proposed algorithm** uses models like **Word2Vec**, **LDA Classifier** and **AdaBoost Classifier** to **detect potential predatory behaviour** with an F1-score of **0.7579 as accuracy**

Conclusion

- Project was the perfect combination of **computational linguistics** and **statistical machine learning**

- The **proposed algorithm** uses models like **Word2Vec**, **LDA Classifier** and **AdaBoost Classifier** to **detect potential predatory behaviour** with an F1-score of **0.7579 as accuracy**

- The **two stage classification system** creates a **3 group classification** of online chat-room conversations

Conclusion

- Project was the perfect combination of **computational linguistics** and **statistical machine learning**

- The **proposed algorithm** uses models like **Word2Vec**, **LDA Classifier** and **AdaBoost Classifier** to **detect potential predatory behaviour** with an F1-score of **0.7579 as accuracy**

- The **two stage classification system** creates a **3 group classification** of online chat-room conversations

- Approach could **enhance children's safety in online environments** by detecting malicious behaviour

Acknowledgements

- Supervisor: Dr. Abdallah Mohamed
- Co-Supervisor: Dr. Jeffrey Andrews
- Family and friends, especially Ryan M., Liam W., Ryan K. and Parsa R.

Thank you.

References:

1. De Boom, Cedric, et al. "Representation learning for very short texts using weighted word embedding aggregation." Pattern Recognition Letters 80 (2016): 150-156.
2. Wolak, J., Finkelhor, D., and Mitchell, K. (2004). Internet-initiated sex crimes against minors: Implications for prevention based on findings from a national study. Journal of Adolescent Health, 35(5):424–e11.
3. Wolak, J., Finkelhor, D., Mitchell, K. J., and Ybarra, M. L. (2008). Online" predators" and their victims: myths, realities, and implications for prevention and treatment. American psychologist, 63(2):111.
4. Dataset can be found at: http://pan.webis.de/clef12/pan12-web/author-identification.html
5. Figure 5 Photo Credits: https://towardsdatascience.com/word-embedding-with-word2vec-and-fasttext-a209c1d3e12c
6. Figure 4 Photo Credits: https://groups.google.com/forum/#!topic/gensim/EwK-6JgkWVI
7. Figure 18 Photo Credits: https://ongxuanhong.wordpress.com/2015/08/25/danh-gia-mo-hinh-model-evaluation/
8. AdaBoost Visualization: https://www.youtube.com/watch?v=BoGNyWW9-mE&t=175s
9. Tibshirani, R., James, G., Witten, D., and Hastie, T. (2013). An introduction to statistical learning-with applications in r.
10. Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.

Appendix

LDA in details

- Z is the linear combination
- Define Score function S(B)

$$Z = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d$$

$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta}$$ Score function

$$\downarrow$$

$$S(\beta) = \frac{\bar{Z}_1 - \bar{Z}_2}{\text{Variance of } Z \text{ within groups}}$$

LDA in details

- Z is the linear combination
- Define Score function S(B)

$$Z = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d$$

$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta}$$ Score function

↓

$$S(\beta) = \frac{\bar{Z}_1 - \bar{Z}_2}{\text{Variance of } Z \text{ within groups}}$$

$$\beta = C^{-1}(\mu_1 - \mu_2)$$ Model coefficients

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 + n_2 C_2)$$ Pooled covariance matrix

Where:

β : Linear model coefficients

C_1, C_2 : Covariance matrices

μ_1, μ_2 : Mean vectors

- Estimate Mean Vectors
- Calculate Covariance Matrices
- Get Model Coefficients

LDA in details

- Z is the linear combination
- Define Score function S(B)

$$Z = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_d x_d$$

$$S(\beta) = \frac{\beta^T \mu_1 - \beta^T \mu_2}{\beta^T C \beta}$$ Score function

$$S(\beta) = \frac{\bar{Z}_1 - \bar{Z}_2}{\text{Variance of } Z \text{ within groups}}$$

$$\beta = C^{-1}(\mu_1 - \mu_2)$$ *Model coefficients*

$$C = \frac{1}{n_1 + n_2}(n_1 C_1 + n_2 C_2)$$ *Pooled covariance matrix*

Where:

β : *Linear model coefficients*

C_1, C_2 : *Covariance matrices*

μ_1, μ_2 : *Mean vectors*

- Estimate Mean Vectors
- Calculate Covariance Matrices
- Get Model Coefficients

$$\beta^T \left(x - \left(\frac{\mu_1 + \mu_2}{2} \right) \right) > \log \frac{p(c_1)}{p(c_2)}$$

Coefficients vector

Data vector

Mean vector

Class probability

A new observation is classified using this equation