Leveraging Support Vector Machines to Automatically Detect Small Talk in Public Service Discourse

Simon Bross
University of Potsdam
bross@uni-potsdam.de

1 Introduction

Even though small talk is an integral part of daily communication, the field of natural language processing has so far paid little attention to automatically detecting it. While at first sight this task might seem to be viable with ease and extraneous outside of academic contexts, it comes along with its intricacies and pertinence to real-life applications.

In chat-bots and virtual assistants, detecting small talk can help to recognize if a user is engaging in casual conversation - as opposed to having a specific query or problem - which can be leveraged to enhance the interaction with naturalness and engagement. Research indicates that small talk is crucial for fostering deeper relationships between virtual agents and their human users, particularly for long-term interactions where building a close connection is essential. This emphasis on relationship-building is evident in the growing interest in "more sociable" agents, such as companion agents (cf. Mattar and Wachsmuth, 2012).

The ability to distinguish small talk from substantive conversation could also prove valuable in automated social media monitoring and content moderation. It could help focus on relevant discussions and improve the accuracy of sentiment analysis. Moreover, in automated surveillance and monitoring systems, differentiating small talk from potentially threatening or relevant conversations could contribute to more effective identification of genuine security concerns.

In the context of Public Service discourse, automatic small talk detection offers several benefits. It allows for improved analysis of interactions, actual concerns, and feedback from citizens without the noise of casual conversation. This enables the identification of key issues and areas needing improvement, ultimately enhancing the quality and effectiveness of public services. Additionally, it provides insight into how engaged public service staff are with citizens. By analyzing the content and frequency of small talk, the level of personal connection staff are establishing can be assessed. High levels of small talk may indicate that staff are taking the time to build rapport and show empathy, while a lack of it might suggest a more transactional approach, potentially indicating time constraints, high workload, or the need for improved communication skills training.

This project aims to develop a Support Vector Machine (SVM) classifier capable of effectively distinguishing between small talk and non-small talk. The methodology employed includes careful dataset selection, preprocessing steps to enhance data quality, hyperparameter tuning of the SVM model to optimize performance, feature scaling to normalize input data, utilization of k-fold cross-validation for robust evaluation, comparison against a baseline model, and a comprehensive error analysis. Various model configurations will be experimented with to identify the most effective approach for this task.

2 Theory on Small Talk

The intricacies of automatically detecting small talk lie in its elusive nature, which complicates both its definition and objective annotation. To address these, the following section will provide a brief overview of the multifaceted nature, functions, and cultural aspects of small talk.

Small talk, also referred to as phatic communication, plays a crucial role in social interactions and has been extensively researched in linguistics, sociology, and psychology due to its significant social functions. Beyond mere 'chit-chat,' small talk serves several important roles: Coupland (2003) argues that it is a vital component of social cohesion, helping to establish and maintain relationships, create a sense of solidarity, and manage interpersonal boundaries. It often occurs at the beginning and end of conversations, serving as a bridge between silence and more substantive dialogue. This bridging role highlights its importance in social rituals and transitions within interactions.

Culturally, the content and structure of small talk can vary significantly across them. Isbister et al. (2000) emphasize that cultural norms heavily influence the topics, duration, and appropriateness of small talk in different contexts. For example, weather-related small talk might be common in some cultures but considered trivial or inappropriate in others. One of the most common forms of small talk is the "How are you?" exchange, which Coupland et al. (1992) describe as a negotiation of "phatic communion". This ritualized interaction serves to acknowledge the other person's presence and establish a baseline of social connection, even if the participants do not engage in deeper conversation.

Mattar and Wachsmuth (2012) further elaborate on the complexity of small talk, noting that it involves more than just exchanging pleasantries. They argue that small talk requires participants to engage in active listening, show empathy, and demonstrate social intelligence. This multifaceted nature of small talk presents challenges for its automatic detection and analysis without taking extralinguistic cues into account. Therefore, capturing the nuances of small talk would require sophisticated models that can interpret both verbal and non-verbal cues.

Furthermore, despite its prevalence and importance, people often underestimate the value of small talk: Epley and Schroeder (2014) found that individuals tend to mistakenly seek solitude in social situations, overlooking the potential benefits of engaging in small talk with strangers. Their research suggests that small talk can lead to increased feelings of well-being and social connection. This finding underscores the psychological benefits of everyday social interactions.

Endrass et al. (2011) further explore the cultural aspects of small talk, emphasizing the need for culturally adaptive systems in human-computer interaction. They argue that virtual agents and conversational AI systems should be capable of engaging in culturally appropriate small talk to enhance user experience and build rapport. This adaptability is essential for creating more natural and engaging interactions between humans and AI.

3 Data and Preprocessing

3.1 Data

The data that this project uses for the automatic detection of small talk derives from the PSE corpus (Espinoza et al., 2024) which collected face-to-face interactions between representatives of the state and citizens in Germany. For his Master's thesis, Steffen Frenzel (2023) had a portion of the corpus' conversational utterances assessed to determine whether they qualify as small talk and used this data to build a classifier based on a logistic regression model. The following definition was established as the foundation for the annotation: Small talk is a conversational practice focused on social bonding rather than exchanging information, typically avoiding controversial topics and requiring no specialized knowledge. It occurs globally across cultures, especially in Western contexts where it helps maintain conversation flow and etiquette. Despite its often negative perception, small talk is recognized as essential for social communication, aiming to preserve face and ease transitions between topics within varied contexts and cultural backgrounds (cf. ibid).

The utterances in the PSE corpus were annotated using a Likert scale ranging from 1 to 6. On this scale, 1 indicates a strong tendency towards the non-small talk class, while 6 indicates a high likelihood of an utterance being small talk. The dataset initially used derived from the second annotation round, in which two annotators performed the small talk rating of the utterances. The values from the two annotators were averaged and the decision boundary was set at 3.5 (i.e. the midpoint of the Likert scale). Scores below 3.5 were classified as non-small talk, whereas scores equal to or above 3.5 were classified as small talk. The dataset from the second annotation round thus showed a significant class imbalance with 2,175 instances labeled as non-small talk and 325 instances labeled as small talk.

Despite efforts to mitigate this imbalance - consisting of oversampling techniques for the minority class and adjusting the SVM classifier's class weight parameter - preliminary evaluation results showed that the classifier performed very poorly in predicting the small talk class, with F1 values amounting to only 0.47. In contrast, the non-small talk class was very well predicted, with F1 scores exceeding 0.9.

As a consequence, the data from the second annotation round was discarded and replaced by the data from the first annotation round. In the first round, only one annotator performed the small talk rating, and only 2,003 utterances were labeled - compared to 2,500 in the second round. By labeling instances as non-small talk if they were rated 3 and below (i.e. 1-3) on the Likert scale and as small talk if rated above 3 (i.e. 4-6), the data revealed a more balanced class distribution with 1,035 instances for non-small talk and 968 instances for small talk. This balance was expected to improve the performance in classifying small talk.

3.2 Data Preprocessing

Given that the textual data contains possible noise in form of annotation (symbols) following the guidlines from Dresing and Pehl (2015), the following preprocessing (in `features.py`) was conducted to remove it:

1. Annotations in round brackets: This includes annotations of pauses and other annotations using round brackets. Example: Pauses like `(.)`, `(..)`, `(...)`, and other annotations like `(unv.)`, `(Axt?)`. Both the brackets and contents are removed, given that they do not provide valuable information for the classification task.

2. Annotations in square brackets: square brackets `[ ]` might be used for marking overlapping speech and other annotations such as anonymized names. Both the brackets and contents are removed as they are not relevant for classification.

3. Slashes and double slashes: Used to denote classification-irrelevant word or sentence breaks and speaker overlaps, respectively, and are therefore removed. It would also be desirable to remove the content preceding the slashes, as this information is typically a redundant and corrected word/sentence attempt. The relevant/corrected information crucial for classification follows the slash. However, this task is challenging due to the need for contextual and grammatical information to determine whether the slash applies solely to the preceding word or the entire sentence. For instance, in the sentence 'Ich habe mir aber Sor/ Gedanken gemacht,' while the word 'Sor' could be removed along with the slash, implementing a general solution is difficult because in another sentence, the slash might be preceded by an entire sentence attempt. It was therefore refrained from removing the contents preceding slashes.

4 Feature Engineering

Thoroughly conducted feature engineering is a critical step in the process of building effective machine learning models. Various linguistic and semantic features were crafted to represent the data in a preferably holistic way that enhances the model's ability to accurately classify small talk, where each feature is designed to capture different aspects of the text. Both traditional techniques like TF-IDF for sparse matrix representations and more modern methods such as dense sentence embeddings for nuanced semantic capture were employed.

4.1 Part-of-Speech (POS) Tagging

In the FeatureExtractor class, POS tagging is performed using SpaCy's German language model `de_core_news_md`. This model is chosen because `nltk` lacks an out-of-the-box POS-tagger for German. The `_pos` method in the class operates on two main input parameters: `training_docs` and `test_docs`, which are lists of SpaCy `Doc` objects. Each `Doc` object represents a processed utterance from the data where each token has been annotated with its respective POS tag (e.g., noun, verb, adjective).

The `_pos` method iterates over the `Doc` objects in both the training and test sets. For each `Doc`, it retrieves the POS tags associated with each tokenized word. These POS tags are then collected into lists, where each list corresponds to an utterance in the dataset. This step ensures that every word in every utterance is tagged with its respective POS tag.

After extracting POS tags, the method employs the `CountVectorizer` from the `sklearn` library to transform the POS tag sequences into count-vectorized matrices. This transformation quantifies the frequency of each POS tag across all utterances in the dataset. The resulting matrices, `matrix_train` and `matrix_test`, represent the syntactic features derived from POS tags for the training and test sets, respectively.

In the context of small talk classification, POS tagging might contribute to capturing syntactic patterns indicative of informal conversation styles. For instance, conversations typically involve frequent use of pronouns, auxiliary verbs, and other functional words, which are reflected in the distribution of POS tags. By incorporating these syntactic features, the model gains insights into the grammatical structure of text, enabling the model to recognize and categorize linguistic patterns that might be indicative of small talk.

4.2 Sentiment

In the FeatureExtractor class, sentiment analysis is integrated using SpaCy's extension SpacyTextBlob, which extends SpaCy's functionalities to include sentiment analysis. This integration allows the model to compute sentiment scores for each utterance, providing a numerical representation of the sentiment polarity ranging from -1 (negative sentiment) to +1 (positive sentiment). When using SpacyTextBlob, the preprocessing of textual data (tokenization and lemmatization) is handled internally by the extension itself. SpacyTextBlob processes each utterance to identify and analyze various linguistic elements such as adjectives, adverbs, and context clues to determine the overall sentiment.

Sentiment scores are expected to contribute to the classification task by capturing the emotional tone and subjective polarity of text, which might be crucial for distinguishing small talk from other types of discourse. In small talk, sentiments are often positive or rather neutral, reflecting casual interactions, whereas negative and neutral sentiments might indicate more serious, non-small talk or formal discourse. However, this is not always the case.

4.3 Complexity

Complexity scores for the utterances are measured using the first variant of the *Wiener Sachtextformel* (Bamberger and Vanecek, 1984) with an implementation from the `textstat` library - a formula originally designed to assess the readability of German texts. This formula calculates a score based on several linguistic features, including sentence length, word length, and the number of syllables per word.

The `textstat` library internally performs several preprocessing steps such as identifying and counting syllables, measuring word lengths, and calculating the average sentence length. This results in a numerical score that reflects the overall complexity of the text. It is computed as follows:

$$\begin{aligned} \text{WSTF1} = {} & 0.1935 \times \text{MS} \\ & + 0.1672 \times \text{SL} \\ & + 0.1297 \times \text{IW} \\ & - 0.0327 \times \text{ES} \\ & - 0.875 \end{aligned} \tag{1}$$

where

- **MS** is the average sentence length in words.
- **SL** is the percentage of words with three or more syllables.
- **IW** is the mean number of syllables per word.
- **ES** is the percentage of words with six or more letters.

Although the Wiener Sachtextformel traditionally assesses text complexity in an educational context, determining the suitability of texts for different reading levels, it can be effectively repurposed for the small talk classification task. Small talk tends to involve simple, straightforward language characterized by shorter sentences and less complex vocabulary. Consequently, texts with lower *Wiener Sachtextformel* scores might tend to be classified as small talk, while higher scores may indicate more formal or complex discourse.

4.4 TF-IDF

TF-IDF (Term Frequency-Inverse Document Frequency) is a fundamental NLP technique to transform textual data into numerical representations, capturing the importance of terms within a corpus. TF-IDF itself is computed as the product of two statistics: term frequency (TF) and inverse document frequency (IDF). Term frequency measures how often a term appears in a document, while inverse document frequency measures how common or rare a term is across the entire corpus. The TF-IDF score for a term t in a document d is computed as follows:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \log\left(\frac{N}{\text{DF}(t)}\right)$$

where $\text{TF}(t, d)$ is the term frequency of term t in document d, N is the total number of documents, and $\text{DF}(t)$ is the number of documents containing term t. This score reflects the importance of a term within a specific document relative to its presence in the entire corpus (cf. Sammut and Webb, 2010, pp. 968-987).

TF-IDF computation is performed by initially tokenizing and lemmatizing the words in the training and test sets using SpaCy. Lemmatization reduces words to their base or dictionary form, helping to normalize the text and reduce dimensionality. After lemmatization, the words are rejoined into sentences, forming the input for the TF-IDF vectorizer.

The TF-IDF vectorizer is instantiated with specific parameters such as `sublinear_tf=True`, `max_df=0.5`, and `stop_words=german_stopwords`. The `sublinear_tf` parameter applies a logarithmic scaling to the term frequency, replacing the raw term frequency with $1 + log(tf)$. This adjustment helps to diminish the effect of very frequent terms, which may not be as informative. By applying logarithmic scaling, the vectorizer ensures that excessively common terms do not overshadow more informative terms (cf Manning et al., 2008, pp. 126-127).

The `max_df` parameter is set to 0.5, meaning that any term appearing in more than 50% of the documents will be ignored. This setting helps in feature reduction by eliminating overly common terms that are likely not useful for the classification task. By setting an upper threshold, `max_df` contributes to feature selection, retaining only those terms that occur in a more balanced proportion of utterances.

In the example code from the seminar on automated small talk detection, TF-IDF was computed on the entire dataset, a practice discouraged in machine learning due to the risk of data leakage. It occurs when information from the test set influences the training process, potentially leading to overly optimistic performance estimates. To mitigate this, in the current project's code, TF-IDF is computed separately for the training and test sets. This separation ensures that the TF-IDF transformation is solely based on the training data, preserving the integrity of the test set for unbiased evaluation.

The vectorizer also uses a predefined list of stop words (`german_stopwords`) to filter out common German words that are generally less useful for classification. Stop words are common words that do not carry significant meaning and are often removed to improve model performance by focusing on more informative terms. However, unlike sentence embeddings that produce dense vector representations capturing semantic meaning in a continuous space, TF-IDF generates sparse matrices (i.e. most of the dimensions with 0 values) which can pose challenges for certain classifiers. The experiments will reveal if this also holds true for a SVM classifier.

By capturing the importance and relevance of terms, TF-IDF helps the model to identify and weigh the terms that contribute most significantly to distinguishing between different types of text. In the context of classifying small talk, TF-IDF can highlight characteristic terms that are indicative of casual, informal communication, as opposed to more formal and non-small talk discourse.

4.5 Sentence Embedding

Sentence embeddings are utilized to represent each utterance as a dense vector in a continuous semantic space, thus capturing the semantic content and context of sentences. The pre-trained model 'paraphrase-multilingual-MiniLM-L12-v2' from the SentenceTransformer library is used, which is specifically designed to encode sentences into dense vector representations (512 dimensions in this case). The `_embedding` method in the FeatureExtractor class takes as input lists of preprocessed training and test utterances. They have undergone basic preprocessing steps to remove unnecessary annotations and symbols, ensuring that the input to the embedding model is cleaner. The SentenceTransformer model is then used to encode these preprocessed utterances into dense embeddings. This process involves mapping each utterance to a vector representation that preserves semantic relationships between utterances. The embedding vectors are computed such that utterances with similar meanings are closer together in the vector space.

For the classification task, these sentence embeddings serve as features that capture the underlying semantic information of each utterance. Unlike sparse representations such as TF-IDF, which focus on the frequency of individual words, sentence embeddings consider the entire context and meaning of the utterance. This representation is expected to enhance the model's ability to classify and generalize. Moreover, using sentence embeddings avoids the sparsity issues inherent in TF-IDF matrices, where most entries are zeros, and thus contributes to a more compact representation.

5 Modelling

5.1 Support Vector Machines

Support Vector Machines (SVM) are a type of supervised machine learning algorithm used for classification and regression tasks that attempt to find the optimal hyperplane (decision boundary) separating the different classes in the data with a maximum margin. A hyperplane is a flat affine subspace of one dimension less than its ambient space; for example, in two-dimensional space, a hyperplane is a line, and in three-dimensional space, a hyperplane is a plane. The margin is the distance between the hyperplane and the closest data points from each class. The key idea is to maximize the margin between the classes, as a larger margin leads to better generalization and avoids overfitting. However, given that most data is not linearly seperable, SVMs map the input data into a higher-dimensional feature space using kernel functions, where the classes become linearly separable. The hyperplane is then constructed in this higher-dimensional space, thus allowing it to handle complex and non-linear relationships within high-dimensional datasets (cf. Weston et al., 2000).

Given that part of the features selected for this project (sentence embeddings and TF-IDF) are high-dimensional, and considering both the strong general performance of SVMs in text classification tasks and their robustness to overfitting, I chose SVMs (implemented by the scikit-learn library) to tackle the task of small talk detection.

5.2 Model Configurations

For this project, various model configurations are considered to investigate their impact on the classification task. The primary objective is to discern how different feature (combinations) and the inclusion of contextual utterances (preceding and following utterance) influence the performance. Specifically, sparse TF-IDF representations will be contrasted with dense sentence embeddings (EMB) - both of which are also further enriched by the syntactic and contextual information provided by part-of-speech tags (POS), sentiment (SENT) , and complexity (COM) scores. The aim consists in identifying the most effective model configuration for accurately distinguishing small talk from non-small talk. The 8 different model configurations are listed in the following:

1. EMB, POS, COM, SENT (no context)
2. EMB, POS, COM, SENT (with context)
3. EMB (no context)
4. EMB (with context)
5. TFIDF, POS, COM, SENT (no context)
6. TFIDF, POS, COM, SENT (with context)
7. TFIDF (no context)
8. TFIDF (with context)

The performance of each configuration will be measured and compared against the baseline model from Frenzel.

5.3 Feature Scaling

Feature scaling is crucial for the SVM classifier to ensure that all features contribute equally to the decision-making process. SVMs are sensitive to the scale of input features because they work by maximizing the margin between classes, which can be disproportionately influenced by features with larger scales. Feature scaling standardizes the range of independent variables (features) so that each feature contributes equally to the distance calculations during the SVM training process. This prevents features with larger numerical ranges from dominating the learning algorithm and allows SVMs to effectively learn the decision boundary without bias towards any particular feature due to its scale.

In the code for this project, the `StandardScaler` from scikit-learn is used to perform the feature scaling. It calculates the mean and the standard deviation of each feature, subsequently adjusting each feature so that the mean value is zero, and the standard deviation is the same for all features. Nonetheless, the relationships between the different features are preserved.

5.4 K-Fold Cross Validation

In the evaluation process of this project, 5-fold cross-validation is employed to assess the performance and robustness of all 8 model configurations. Cross-validation is a fundamental technique in machine learning used to evaluate the generalizability and robustness of a model. It partitions the dataset into k (here: 5) subsets of approximately equal size. In each iteration of the cross-validation process, one of these subsets serves as the test set, while the remaining (here: 4) subsets are used to train the model. This procedure is repeated k times, with each subset taking turns as the test set.

The primary advantage of cross-validation lies in its ability to provide a more reliable estimate of the model's performance compared to a single train-test split. By averaging the performance metrics across multiple folds, cross-validation reduces the variance of the evaluation results and provides a more accurate assessment of how well the model will generalize to unseen data. It helps in identifying and mitigating issues such as overfitting or underfitting by ensuring that the model's performance is evaluated on multiple, independent subsets of data.

The evaluation metrics from the scikit-learn classification report are computed and averaged across all folds. The use of macro-averaging - where the metrics precision, recall, and F1-score are averaged across classes without considering class imbalance - was chosen to provide an unbiased assessment of each model configuration's performance. Furthermore, the baseline model established by Steffen Frenzel (logistic regression model with contextual utterances) serves as a reference point for comparison. This practice is crucial in machine learning experiments as it provides a benchmark against another model, ascertaining whether another model (configuration) contributes to improved classification.

5.5 Hyperparameter Finetuning

After 5-fold cross-validation, the best and worst performing models are identified based on their macro-averaged F1 scores. Subsequently, hyperparameter fine-tuning is conducted on these models aiming to optimize their performance.

The purpose of hyperparameter fine-tuning is crucial as it aims to identify the optimal combination of model parameters that maximize predictive performance on unseen data. Fine-tuning is performed using GridSearchCV, which systematically explores various hyperparameter combinations to find the configuration that yields the highest macro-averaged F1-score. Unlike accuracy, which is often the default metric for GridSearchCV, F1-score was specifically chosen here because it balances both precision and recall, providing a more comprehensive evaluation of the model's performance in classification tasks.

The search space (grid) used for hyperparameter fine-tuning is defined as follows:

1. C: [0.1, 0.5, 1, 2, 3]. The parameter C controls the regularization of the SVM model. It balances the trade-off between achieving a low error on the training data and minimizing the complexity of the model. A small value of C makes the decision surface smooth, while a large value of C aims to classify all training examples correctly.
2. kernel: ['poly', 'rbf', 'sigmoid']. The kernel parameter specifies the kernel type to be used in the SVM algorithm. The kernel function in SVMs transforms the input data into a higher-dimensional space to make it possible to find a linear separation between classes that are not linearly separable in the original feature space. The options used are:
   - 'poly': Polynomial kernel, useful for non-linear data.
   - 'rbf': Radial Basis Function kernel, effective in high-dimensional spaces.
   - 'sigmoid': Sigmoid kernel, resembling a neural network's activation function.
3. gamma: ['scale', 'auto', 1, 0.1, 0.01, 0.001, 0.0001]. The gamma parameter defines how far the influence of a single training example reaches, with low values meaning far and high values meaning close.
4. class_weight: ['balanced', None]. The class_weight parameter is used to assign weights to different classes. It is particularly useful for handling imbalanced datasets - which is the case for the data from the second annotation round.

GridSearchCV is initialized with the SVM estimator, the predefined parameter grid, and a 5-fold cross-validation strategy using StratifiedKFold. The grid search process is then executed by fitting the SVM model to the training data. GridSearchCV systematically evaluates each combination of hyperparameters defined in the grid. For each combination, the model is trained and validated across the five folds, and the macro-averaged F1-score is calculated for each fold.

Once all hyperparameter combinations have been evaluated, GridSearchCV identifies the best combination that yields the highest macro-averaged F1-score. The best parameters are then used to instantiate the final SVM model. Finally, the best estimator from the grid search is used to perform further evaluation. The run_experiment function is called to conduct 5-fold cross-validation and evaluation using the tuned hyperparameters. This additional evaluation ensures that the model's performance is thoroughly assessed and validated.

6 Results

Feature(s)	Finetuned	Context	P_0	P_1	R_0	R_1	F_0	F_1	ACC	P_MAVG	R_MAVG	F_MAVG
Baseline			**0.71**	**0.69**	**0.75**	**0.65**	**0.73**	**0.67**	**0.7**	**0.7**	**0.7**	**0.7**
EMB, POS, COM, SENT	**No**	**Yes**	**0.73**	**0.72**	**0.74**	**0.7**	**0.73**	**0.71**	**0.72**	**0.72**	**0.72**	**0.72**
EMB, POS, COM, SENT	No	No	0.72	0.71	0.73	0.69	0.72	0.7	0.71	0.71	0.71	0.71
EMB	No	Yes	0.7	0.72	0.74	0.7	0.73	0.71	0.72	0.72	0.72	0.72
EMB	No	No	0.71	0,7	0.73	0.69	0.72	0.69	0.71	0.71	0.71	0.71
TFIDF, POS, COM, SENT	No	Yes	0.62	0.63	0.68	0.58	0.66	0.6	0.63	0.63	0.63	0.63
TFIDF, POS, COM, SENT	No	No	0.6	0.63	0.73	0.48	0.66	0.55	0.61	0.62	0.61	0.6
TFIDF	No	Yes	0.63	0.63	0.69	0.58	0.66	0.6	0.63	0.63	0.63	0.63
TFIDF	**No**	**No**	**0.6**	**0.63**	**0.74**	**0.47**	**0.66**	**0.54**	**0.61**	**0.62**	**0.61**	**0.6**
EMB, POS, COM, SENT	**Yes**	**Yes**	**0.73**	**0.71**	**0.73**	**0.72**	**0.73**	**0.72**	**0.72**	**0.72**	**0.72**	**0.72**
TFIDF, POS, COM, SENT	**Yes**	**No**	**0.63**	**0.61**	**0.65**	**0.59**	**0.64**	**0.6**	**0.62**	**0.62**	**0.62**	**0.62**

Table 1: Averaged experiment results across all folds and combinations of features and context. EMB = sentence embeddings, POS = POS-tags, COM = complexity, SENT = sentiment. The table provides class-specific metric values (ending with _0, _1) for Precision (P), Recall (R), and F1-score (F), as well as overall metrics including Accuracy (ACC) and macro-averaged Precision, Recall, and F1-score. The baseline performance (LogReg with context, results from (Frenzel, 2023) is presented in the first row. The penultimate and last rows display the performance of the best and worst model configurations, respectively, after hyperparameter tuning.

The baseline from Frenzel (2023) using logistic regression with contextual utterances demonstrated an overall accuracy, macro averaged precision, recall, and F1 of 0.70, respectively. Precision, recall, and F1-score scores for class 0 (non-small talk) amount to 0.71, 0.75, and 0.73, while class 1 (small talk) showed slightly lower values at 0.69, 0.65, and 0.67. These results indicate that the baseline model performs reasonably well, with balanced performance across both classes, yet with a marginally better performance in detecting non-small talk.

When the combination of sentence embeddings, POS-tags, complexity, and sentiment features (EMB, POS, COM, SENT) was used with contextual utterances, the model's performance improved slightly compared to the baseline. The overall accuracy, macro averaged precision, recall, and F1 increased to 0.72, and both classes exhibited better-balanced metric values, with class 0 and class 1 achieving precision, recall, and F1-scores between 0.7 and 0.74. When contextual utterances were removed, the performance dropped marginally to an accuracy of 0.71.

Using only sentence embeddings with context resulted in an overall accuracy, macro averaged precision, recall, and F1 of 0.72, matching the performance when all features were used.. The Precision, recall, and F1-scores for both classes remained rather similar to the EMB, POS, COM, SENT configuration with context, indicating that embeddings alone can capture significant information for small talk detection. Without context, the performance slightly declined by 0.01 points for macro averaged precision, recall and F1.

In contrast, using TFIDF combined with POS-tags, complexity, and sentiment showed a notable drop in performance. With context, the overall accuracy, macro averaged precision, recall, and F1 plummeted to 0.63, and the metric values for both classes decreased significantly, with precision, recall, and F1-scores 0.58 and 0.68 when using context, and between 0.48 and 0.73 without context. Without context, the performance of this configuration further declined to an accuracy of 0.61, macro averaged precision of 0.62, macro averaged recall of 0.61 and macro averaged F1 of 0.6. With class 1 metrics particularly affected, the results indicate that TFIDF is less effective than sentence embeddings for this classifier. However, also for TFIDF, the additional context from POS, COM, and SENT does not account for a significant performance difference, given that the performance of the TFIDF-only configuration performs equally well as the configuration with the additional features.

Finetuning the best-performing model (EMB, POS, COM, SENT with context) yielded stable results, maintaining an accuracy of 0.72. The metrics for both classes show both minor improvements and decline, indicating that the model was already well-optimized. However, the worst performing model (TF-IDF, POS, COM, SENT without context) showed some notable changes after finetuning: The accuracy increased marginally to 0.62, and precision for class 0 improved from 0.60 to 0.63, while precision for class 1 remained steady at 0.61. Yet there was a decrease in recall for class 0, dropping from 0.73 to 0.65, whereas recall for class 1 improved from 0.48 to 0.59. F1-score for class 0 remained unchanged at 0.64 and F1-score for class 1 increased from 0.55 to 0.60. These results highlight the nuanced impact of finetuning on model performance metrics. While finetuning led to modest improvements in accuracy and precision for class 0, it also resulted in trade-offs such as reduced recall for class 0 but improved recall and F1-score for class 1. This demonstrates the complex interaction between model adjustments through finetuning and the initial quality and composition of the feature set used.

7 Error Analysis

Although the employed metrics are essential to measuring the performance, correctness, and quality of a system, they are only an approximation of the system's qualities as they are limited to their underlying criteria. The following error analysis aims to counteract the metrics' biases and drawbacks, namely by uncovering error patterns that the metrics hide under the guise of numerical values. The confusion matrices (cf. Appendix)of the two finetuned models (which were both the best and worst performing before tuning) will be analyzed and compared.

The best performing model's confusion matrix reveals 146 true negatives (TN) and 135 true positives (TP) for class 0 (non-small talk, 207 instances in total) and class 1 (small talk, 194 instances in total), respectively. It also shows 61 false positives (FP) and 59 false negatives (FN). These errors indicate instances where non-small talk was incorrectly classified as small talk and vice versa. Overall, this model achieved a higher accuracy and balanced precision-recall trade-off compared to the worst performing model.

Conversely, the worst performing model's confusion matrix displays 135 true negatives and 103 true positives, but with 72 false positives and 91 false negatives. This model has only 12 fewer true negatives but 32 fewer true positives compared to the best performing model. The larger difference in true positives (32) suggests that the worst performing model after finetuning is significantly less effective at correctly identifying small talk instances, while the smaller difference in true negatives (12) indicates a somewhat more similar capability in correctly identifying non-small talk instances.

The worst performing model exhibits a higher rate of misclassifications compared to the best performing, particularly in the form of false positives and false negatives. Notably, the worst performing model tends to misclassify small talk as non-small talk more frequently, which is evident from the higher number of false negatives (91). This means that genuine small talk instances are often missed. On the other hand, the best performing model, although generally more accurate, exhibits a slight tendency to misclassify non-small talk as small talk, as indicated by its 61 false positives versus 59 false negatives.

8 Discussion

Despite efforts in hyperparameter tuning and incorporating additional information from SENT, COM, and POS, improvements in model performance and exceeding the baseline performance were achieved but limited. In general, models incorporating dense sentence embeddings consistently outperformed those using TF-IDF representations. This was already expected during the feature engineering phase, which anticipated that some classifiers would struggle with sparse TF-IDF representations. The results provide evidence for this, as the most effective configuration was sentence embeddings enriched with POS tags, complexity, sentiment, and contextual utterances. This outcome suggests that dense embeddings are highly effective in capturing the nuanced information necessary for differentiating small talk from non-small talk. However, the results also show that the additional context from POS, COM, and SENT and contextual utterances only adds a possibly negligible value to the classifier, yet not improving it considerably.

In stark contrast, models that relied on TF-IDF-based features, even when enriched with additional contextual or linguistic information, consistently underperformed. Dense embeddings' advantage over TF-IDF probably lies in mitigating matrix sparsity issues inherent in TF-IDF representations, where SVMs could struggle with high-dimensional sparse data due to irrelevant or noisy dimensions hindering effective class separation.

The process of fine-tuning the best-performing model maintained stable results, reinforcing the robustness of the chosen features and model configuration. However, fine-tuning the worst-performing model, while slightly improving accuracy, revealed significant trade-offs between precision and recall for the two classes. This indicates that while fine-tuning can enhance model performance, its benefits are highly dependent on the initial feature set and model configuration. This finding emphasizes the need for careful feature selection and model optimization to achieve optimal results.

The error analysis further illuminated the strengths and weaknesses of the model configurations. The best performing model's confusion matrix revealed a balanced distribution of true positives and true negatives, with fewer misclassifications compared to the worst performing model. Specifically, the best performing model showed 146 true negatives and 135 true positives for non-small talk and small talk, respectively, with 61 false positives and 59 false negatives. This balanced error distribution indicates the model's robustness in correctly identifying both classes. Conversely, the worst performing model exhibited a higher rate of misclassifications, particularly in the form of false negatives (91) and false positives (72), indicating its struggles in correctly classifying small talk instances. The implications of these misclassifications depend on the context/application in which the classifier is embedded. False positives, where non-small talk is incorrectly identified as small talk, can lead to the omission of significant information in automated analysis. In the context of Public Service discourse, this could result in important issues being overlooked, affecting the analysis' quality. On the other hand, if genuine small talk is misclassified as non-small talk, it could lead to a misinterpretation of the conversational dynamics, such as failing to recognize the intended casual or informal tone of the interaction. This misclassification could result in inappropriate responses or actions based on the misinterpreted context, which may be critical in applications ranging from customer service and social media monitoring to automated content moderation and sentiment analysis.

Several challenges were encountered during the project. Firstly, issues arose from the dataset, particularly from the second annotation round. This dataset exhibited a notable class imbalance, with a majority of instances labeled as non-small talk and very few as small talk. Despite attempts to address this imbalance using oversampling and applying class weights, the SVM model struggled to effectively predict the small talk class, resulting in sub-optimal F1 scores. However, it remains possible that alternative classifiers could potentially handle this imbalance more effectively than SVMs.

Additionally, preprocessing the textual data posed a challenge which was primarily due to the presence of certain annotations that are hard to remove correctly. Despite efforts to systematically remove them to enhance data quality, difficulties persisted in accurately determining and removing content preceding slashes, and it was ultimately refrained from.

Nevertheless, despite these challenges, the eventual evaluation results showed significant improvements. By discarding the problematic dataset from the second annotation round and focusing on the more balanced data from the first round, the performance in classifying small talk considerably improved. This underscored the importance of data quality and distribution in achieving robust classification outcomes.

In order to further improve the results in small talk classification, several key enhancements could be implemented. Firstly, expanding the dataset size would be beneficial, ideally ensuring it is balanced and annotated by multiple annotators. A larger dataset allows for better representation of the variability and complexity inherent in conversational data, reducing the risk of overfitting and improving generalization to unseen examples. Annotations from multiple annotators provide a broader perspective on the classification task and enhance the reliability and robustness of the labeled data.

Secondly, refining the preprocessing steps - particularly the annotation removal - could be beneficial. Enhancing the preprocessing to accurately identify and remove irrelevant annotations while preserving contextually relevant information is essential for improving data cleanliness and subsequently enhancing model performance.

Thirdly, feature selection, especially for TF-IDF representations, is crucial when using SVMs. SVMs perform optimally with a reduced set of informative features, as excessive features can lead to increased computational complexity and potential overfitting. By employing more profound feature selection techniques, irrelevant features can be filtered out, focusing on those most discriminative for the classification task. This not only enhances the efficiency of the SVM model but also improves its ability to generalize to new data.

Lastly, exploring alternative classifiers beyond SVMs, including neural networks and ensemble methods, could yield further performance gains. Neural networks in particular can capture complex nonlinear relationships in data, which may be beneficial given the nuances and variability in conversational language. Experimenting with different architectures and configurations of neural networks, alongside traditional classifiers, allows for a comprehensive comparison of their strengths and weaknesses in small talk classification. Future research in small talk detection could also focus on developing more sophisticated models that account for cultural variations, context-awareness, and the dynamic nature of social interactions as outlined in the theory section.

9 Conclusion

This project aimed to explore the automatic detection of small talk, presenting both challenges and opportunities for enhancing human-computer interactions and understanding social dynamics within Public Service discourse. Various machine learning techniques were employed and experimented with to address this task, with the core objective to identify the most effective approach for distinguishing small talk from non-small talk using SVMs.

The experiments revealed that the performance of model configurations using dense sentence embeddings constantly exceeded those employing TF-IDF representations in differentiating small talk from non-small talk. The most effective configuration included a combination of sentence embeddings, POS tags, complexity measures, sentiment scores, and contextual utterances. However, the marginal improvement over the baseline model underscores the inherent complexity of detecting small talk and the necessity for further experimentation.

Several limitations were encountered during this project, including preprocessing difficulties and imbalanced dataset, the latter highly restricting the model's ability to predict small talk instances effectively. Furthermore, the additional features (POS tags, complexity measures, and sentiment) that were expected to provide valuable information for the classification did not significantly enhance performance beyond what was achieved with sentence embeddings alone. Including contextual utterances These limitations suggest that more sophisticated feature engineering and alternative modeling approaches are necessary to capture the nuances of small talk accurately.

Despite these challenges, the experiments provided valuable insights into the complexities of small talk detection and highlighted areas for improvement. To improve performance, future experiments should focus on advanced embedding techniques or hybrid models that combine the strengths of different representations. Additionally, more sophisticated feature engineering methods or alternative machine learning models, such as neural networks, could better capture the subtle nuances of small talk. Addressing the data issue by expanding the dataset - preferably in a balanced fashion - and ensuring comprehensive annotation can further enhance the model's ability to generalize. Future research could also consider integrating extra-linguistic knowledge or even multimodal data, such as audio and visual cues, to provide a richer context for small talk detection. This could lead to more accurate and robust models by capturing non-verbal cues that are essential in understanding social dynamics.

In conclusion, while this project has made incremental progress in the automatic detection of small talk, it has also illuminated the inherent complexities of this task, resulting from the elusive nature of small talk itself. The findings underscore the importance of continued research and innovation to improve the ability to detect and analyze small talk. Enhancing human-computer interaction and understanding social communication dynamics in Public Service discourse has significant implications, and addressing the limitations identified in this project will be pivotal.

References

R. Bamberger and E. Vanecek. 1984. *Lesen, Verstehen, Lernen, Schreiben : die Schwierigkeitsstufen von Texten in deutscher Sprache*. Wien [u.a.].

Justine Coupland. 2003. Small talk: Social functions. *Research on Language and Social Interaction*, 36(1):1–6.

Justine Coupland, Nikolas Coupland, and Jeffrey D. Robinson. 1992. "How Are You?": Negotiating Phatic Communion. *Language in Society*, 21(2):207–230.

T. Dresing and T. Pehl. 2015. *Praxisbuch Interview, Transkription & Analyse: Anleitungen und Regelsysteme für qualitativ Forschende*. Dresing.

Birgit Endrass, Yukiko Nakano, Afia Akhter Lipi, Matthias Rehm, and Elisabeth André. 2011. Culture-Related Topic Selection in Small Talk Conversations across Germany and Japan. In *Intelligent Virtual Agents*, pages 1–13, Berlin, Heidelberg. Springer Berlin Heidelberg.

Nicholas Epley and Juliana Schroeder. 2014. Mistakenly seeking solitude. *Journal of Experimental Psychology: General*, 143(5):1980–99.

Ingrid Espinoza, Steffen Frenzel, Laurin Friedrich, Wassiliki Siskou, Steffen Eckhard, and Annette Hautli-Janisz. 2024. PSE v1.0: The First Open Access Corpus of Public Service Encounters. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 13315–13320, Torino, Italia. ELRA and ICCL.

Steffen Frenzel. 2023. Automatic Detection of Small Talk in German Public Service Encounters. Master's thesis.

Katherine Isbister, Hideyuki Nakanishi, Toru Ishida, and Clifford Nass. 2000. Helper Agent: Designing an Assistant for Human-Human Interaction in a Virtual Meeting Space. *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, Cambridge, UK.

Nikita Mattar and Ipke Wachsmuth. 2012. Small Talk Is More than Chit-Chat. In *KI 2012: Advances in Artificial Intelligence*, pages 119–130, Berlin, Heidelberg. Springer Berlin Heidelberg.

Claude Sammut and Geoffrey I. Webb, editors. 2010. *Encyclopedia of Machine Learning*. Springer.

Jason Weston, Sayan Mukherjee, Olivier Chapelle, Massimiliano Pontil, Tomaso Poggio, and Vladimir Vapnik. 2000. Feature Selection for SVMs. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press.

A Appendix

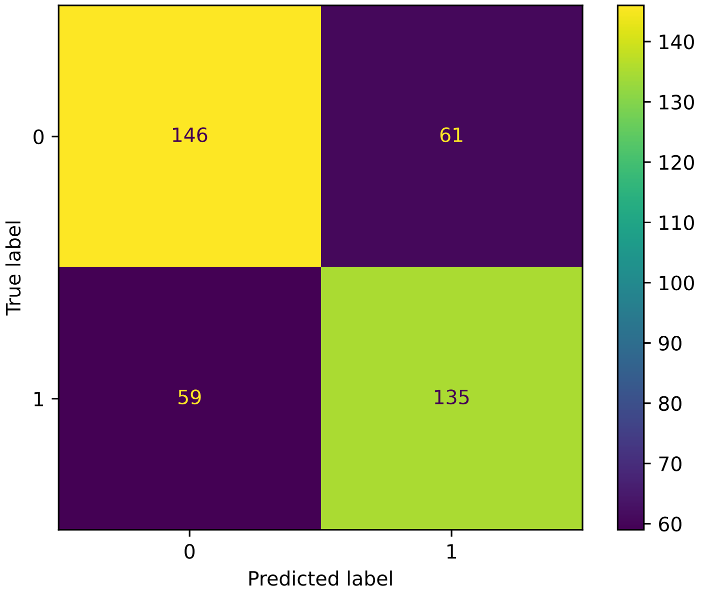

Figure 1: Confusion matrix for the finetuned model (EMB, POS, COMP, SENT, with context) that performed best before tuning, fold 1.

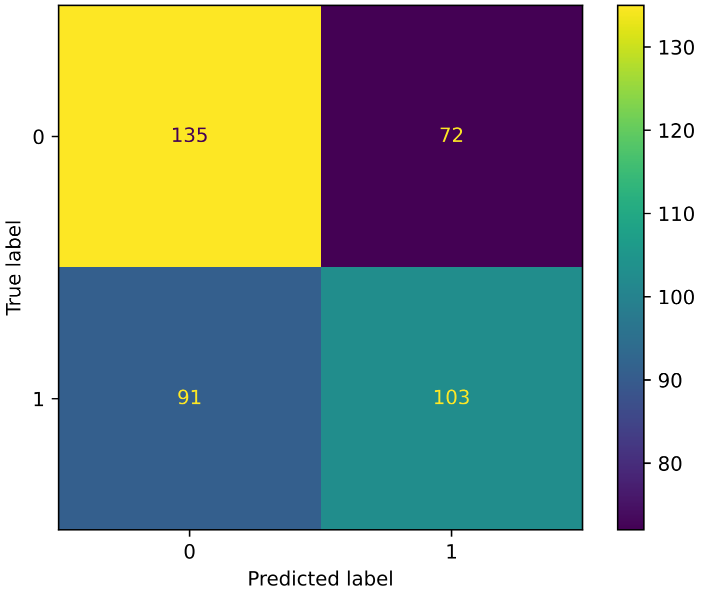

Figure 2: Confusion matrix for the finetuned model (TFIDF, POS, COMP, SENT, without context) that performed worst before tuning, fold 1.