MARMARA UNIVERSITY
INSTITUTE FOR GRADUATE STUDIES
IN PURE AND APPLIED SCIENCES

NAMED ENTITY RECOGNITION FOR E-COMMERCE SEARCH QUERIES IN TURKISH

BEYZANUR SARAÇLAR

MASTER THESIS

Department of Computer Engineering (English)
Data Science Program

Thesis Supervisor

Assoc. Prof. Dr. Mustafa AĞAOĞLU

ISTANBUL, 2024

TABLE OF CONTENTS

1. INTRODUCTION 1
1.1. Overview of Natural Language Processing (NLP) Models 1
1.1.1. Definition of Natural Language Processing 1
1.1.2. Development of NLP Models 1
1.1.3. Importance of NLP in Various Applications 3
1.2. Development of NLP Models in Named Entity Recognition 5
1.2.1. Definition of Named Entity Recognition (NER) and Focus on NER Applications 5
1.2.2. Notable NLP Architectures for NER 6
1.2.3. Comparative Analysis of NER Models 8
1.3. Rationale for the Need of NER in E-commerce Queries 11
1.3.1. Challenges in E-commerce Search Queries 11
1.3.2. Role of NER in Enhancing Search Query Processing 12
1.3.3. Significance of NER in Turkish Language Processing 12
1.3.4. Existing NER Models in E-commerce Query 14
1.4. Aim of the Thesis 16
1.4.1. Objectives of the Research and Contribution to the Field of NLP and E-commerce 16
2. MATERIALS AND METHODS 17
2.1. Dataset 17
2.1.1. Dataset Collection and Pre-processing 17
2.1.2. Annotation Process 18
2.1.3. Dataset Statistics 19
2.2. Model Architecture 20
2.2.1. Baseline Models 21
2.2.2. Transformers-Based Models 26
2.3. Experimental Setup 28
2.3.1. Vectorization of Text Data 28
2.3.2. Hyperparameter Tuning and Training of the Models 29
2.4. Evaluation Metrics 31
3. RESULTS AND DISCUSSION 33
3.1. Experimental Results 33
3.1.1. Baseline Models 33
3.1.2. Baseline Models with Word Embeddings 34
3.1.3. Computational Resources Comparison of the Baseline Models 35
3.1.4. Transformers-Based Models 36
4. CONCLUSION 40

ÖZET

TÜRKÇE E-TİCARET ARAMA SORGULARI İÇİN ADLANDIRILMIŞ VARLIK TANIMA

E-ticaretin gelişmesi, çevrimiçi işlemlerin sayısında hızlı bir artışa yol açarak, arama motorlarını tüketicilerin ürün ve hizmetleri bulmasında önemli bir araç haline getirdi. Adlandırılmış Varlık Tanıma'nın (NER) kullanıcı sorgularına uygulanması, e-ticaret platformlarının ürün keşfi ve kullanıcı deneyimini geliştirmek için ağırlıklı olarak kullanıcı arama sorgularına dayanması nedeniyle özellikle önemli hale gelmektedir. Bu araştırma, Türkçe dilinde e-ticaret arama sorgularına NER yönteminin uygulanmasına daklanmaktadır. Bu amaçla Türkçe dilinde açıklamalı bir e-ticaret arama sorgusu veri kümesi oluşturduk. Çalışmada ön eğitimli modellere ince ayar yaparak varlık tanımada yüksek performans elde etmek için kelime yerleştirme ve dönüştürücü modeller kullanılmıştır. Önceden eğitilmiş modeller, eğitim aşamalarında devasa derlemler üzerinde kodlanan dil bilgisinden yararlanarak dilin yapısı ve bağlamsal özellikleri sunması açısından önemli bir avantaj sunar. Bu yaklaşım, NER sisteminin, alana özgü kapsamlı eğitim verileri gerektirmeden, e-ticaret Türkçe arama sorgularına özgü bağlamsal ve dilsel özellikleri kavramasını sağlar.

Sonuçlara göre tüm Transformers tabanlı modeller, tüm ölçümlerde temel modellerden daha iyi performans gösteriyor ve bu da büyük miktarda veriyle ön eğitimin üstün performansını gösteriyor. Bireysel transformatör modelleri arasında ELECTRA %91,97 ve %84,27 ile sırasıyla en yüksek wegihted ve ortalama macro F1-score'une elde etmiştir. Öte yandan ön eğitimli BERT word ebedding amacıyla Bi-LSTM+CRF modeli ile kombinasyonu, tüm modeller arasında %92,49 ve %84,34 ile sırasıyla en yüksek wegihted ve ortalama macro F1-score'une elde etmiştir. Ayrıca yapılan deneyler göstermektedir ki kaynakların kısıtlı olduğu ortamlar için DistilBERT ve ConvBERT, performans ve verimlilik arasında dengeli bir uzlaşma sunabilirken, maksimum performans gerektiren görevlerde, daha yüksek kaynak gereksinimlerine rağmen BERT+Bi-LSTM+CRF ve ELECTRA tercih edilebilir. Buna ilaveten BERT modelinin tek başına performans ve verimlilik arasında iyi bir denge kurduğu gözlemlenmiştir.

ABSTRACT

NAMED ENTITY RECOGNITION FOR E-COMMERCE SEARCH QUERIES IN TURKISH

The progress in e-commerce has led to a rapid rise in online transactions, making search engines an essential tool for consumers searching for products and services. The application of Named Entity Recognition (NER) to user queries has become particularly important for e-commerce platforms, as they heavily rely on user search queries to enhance product discovery and user experience. This research focuses on the application of NER methods to e-commerce search queries in the Turkish language. To this end, we have created an annotated dataset of e-commerce search queries in Turkish. In the study, word embeddings and transformer models were used to achieve high performance in entity recognition by fine-tuning pre-trained models. The use of pre-trained models offers a substantial advantage through the utilization of language structure and contextual features encoded in massive corpora during training stages. This approach enables the NER system to grasp the contextual and linguistic characteristics specific to Turkish e-commerce search queries without requiring extensive domain-specific training data.

According to the results, all transformer-based models outperform baseline models across all metrics, demonstrating the superior performance of pre-training with large amounts of data. Among individual transformer models, ELECTRA achieved the highest weighted and macro avg. F1-scores with 91.97% and 84.27%, respectively. However, the combination of the pre-trained BERT model with the Bi-LSTM+CRF model for word embeddings achieved the highest weighted and macro avg. F1-scores among all models, with 92.49% and 84.34%, respectively. Additionally, experiments indicate that in resource-constrained environments, DistilBERT and ConvBERT offer a balanced trade-off between performance and efficiency, while for tasks requiring maximum performance, BERT+Bi-LSTM+CRF and ELECTRA may be preferred despite their higher resource requirements. Furthermore, BERT alone was observed to strike a good balance between performance and efficiency.

ABBREVIATIONS

AI : Artificial Intelligence

BERT : Bidirectional Encoder Representations from Transformers

Bi-LSTM : Bidirectional Long Short-Term Memory

CBOW : Continuous Bag of Words

CLIP : Contrastive Language-Image Pretraining

CRF : Conditional Random Fields

DistilBERT: Distilled Bidirectional Encoder Representations from Transformers

ELECTRA : Efficiently Learning an Encoder that Classifies Token Replacements Accurately

GloVe : Global Vectors

GPT : Generative Pre-trained Transformer

GRU : Gated Recurrent Units

HMM : Hidden Markov Model

LSTM : Long Short-Term Memory

MEM : Maximum Entropy Models

NER : Named Entity Recognition

NLP : Natural Language Processing

RNN : Recurrent Neural Network

SVM : Support Vector Machines

LIST OF FIGURES

FIGURE 2.1. LABEL FREQUENCIES OF THE BIO FORMATTED DATASET AS PERCENTAGES. 19

FIGURE 2.2. OVERVIEW OF THE PROPOSED METHOD. 21

FIGURE 2.3. LSTM RECURRENT UNIT. 22

FIGURE 2.4. BI-LSTM+CRF MODEL ARCHITECTURE. 24

FIGURE 2.5. BERT+BILSTM+CRF MODEL ARCHITECTURE ILLUSTRATION. 28

FIGURE 3.1. WEIGHTED AND MACRO AVG. F1-SCORES OF THE BI-LSTM-BASED MODELS. 35

FIGURE 3.2. PERFORMANCE OF THE TRANSFORMERS-BASED MODEL ACCORDING TO THE PRECISION, RECALL, WEIGHTED AND MACRO AVG. F1-SCORE. 37

FIGURE 3.3. RESOURCE USAGE OF THE TRANSFORMERS-BASED MODELS. 37

LIST OF TABLES

TABLE 2.1. BIO TAGGING FORMAT EXAMPLE. 18

TABLE 2.2. DATASET STATISTICS. 20

TABLE 2.3. TRAINING PARAMETERS OF THE EMBEDDING MODELS. 30

TABLE 3.1. PERFORMANCE OF THE BASELINE MODELS ACCORDING TO THE ACCURACY, PRECISION, RECALL, WEIGHTED F1-SCORE, MACRO AVG. F1-SCORE. 33

TABLE 3.2. PERFORMANCE OF THE BASELINE MODELS WITH DIFFERENT WORD EMBEDDING MODELS ACCORDING TO THE ACCURACY, PRECISION, RECALL, WEIGHTED F1-SCORE AND MACRO AVG. F1-SCORE. 34

TABLE 3.3. COMPUTATIONAL RESOURCES COMPARISON OF THE BASELINE MODELS ACCORDING TO THE RAM USAGE, AVERAGE TIME FOR 1 EPOCH AND GPU/CPU USAGE. 36

TABLE 3.4. PERFORMANCE OF THE TRANSFORMERS-BASED MODELS ACCORDING TO THE ACCURACY, PRECISION, RECALL, WEIGHTED F1-SCORE, MACRO AVG. F1-SCORE AND RAM USAGE. 36

TABLE 3.5. PERFORMANCE OF THE BERT MODEL ACCORDING TO THE ACCURACY, PRECISION, RECALL, F1-SCORE FOR EACH ENTITY. 39

1. INTRODUCTION

1.1. Overview of Natural Language Processing (NLP) Models

1.1.1. Definition of Natural Language Processing

Natural Language Processing is a discipline within Artificial Intelligence (AI) centered on facilitating human-computer interactions using natural language. The fundamental purpose of NLP is to enable machines to comprehend, process and generate human language in a way that is meaningful and fits the context. [1]. This field encompasses various tasks, covering areas such as speech recognition, language interpretation, text generation, and translation.

NLP employs a combination of computational linguistics, computer science, and machine learning techniques to manage and interpret large-scale natural language data. The aim is to allow computers to comprehend and respond to human language in a way that mirrors human-to-human communication. NLP has applications in various fields, such as sentiment analysis, named entity recognition, machine translation, text generation, text summarization, and more.

To tackle these challenges, NLP employs a variety of techniques, from traditional rule-based systems and statistical models to cutting-edge deep learning methods like recurrent neural networks (RNNs) and transformer models. With technological progress, NLP is continually advancing, expanding the capabilities of machines in understanding and processing human language.

1.1.2. Development of NLP Models

NLP methods initially emerged from fundamental linguistic theories and constructing rule-based systems. During this period, researchers focused on acquiring linguistic knowledge and establishing language rules through computational models [2]. Subsequently, there was a notable shift toward statistical models, leveraging large corpora of text to train probabilistic models for various language tasks. These models allowed for learning complex language patterns, leading to improved performance in tasks like part-of-speech tagging and machine translation. The superiority of machine learning techniques resulted in the widespread adoption of models such as Hidden Markov Models [3], Maximum Entropy [4], Conditional Random Fields [5], and Clustering techniques, representing a significant advancement in the development of NLP. In addition, the development of large text corpora facilitated the training of models that could capture statistical regularities in language [6].

Over the past decade, the domain has experienced notable evolution precipitated by the advent of deep learning. The introduction of neural networks has contributed to notable breakthroughs and expanded the frontiers of research possibilities in the field, played a crucial role in this transformation [7], [8], [9], [10], [11], [12]. Notably, recurrent neural networks (RNNs). RNNs are structured to manage sequential data by capturing temporal dependencies within linguistic contexts. An RNN processes each step of a sequence by taking an input, generating an output, and updating its hidden state. The current input and the previous hidden state impact the memory. This hidden state, often considered a network memory, is influenced by both the current input and the information stored in the hidden state from the previous step. This recurrent mechanism allows RNNs to identify dependencies and patterns in sequential data, making them particularly effective for scenarios where context is critical. RNNs addressed the constraints of feedforward networks by incorporating recurrent connections, enabling the retention of contextual information over extended sequences [13].

Despite their capability to model sequential dependencies, traditional RNNs have limitations, particularly in handling long-term dependencies. To tackle this problem, variations of RNNs such as Long Short Term Memory (LSTM) [14] and Gated Recurrent Unit (GRU) [15] architectures have been developed. Their improved gate mechanism demonstrated high performance in tasks that involve sequential analysis, such as language modeling, machine translation, forecasting and classification tasks. The works of [16], [17] and [18] demonstrated notable performance of these architectures, particularly in the domain of machine translation. In the context of text classification, the studies conducted by [19] and [20] illustrated the effectiveness of LSTM-based models.

Furthermore, the period under consideration observed the advantage of word embeddings as integral components within the domain of NLP. Significant implementations, including Word2Vec [21], FastText [22] and GloVe [23] gained widespread recognition for their ability to identify contextual connections between words. By analyzing word usage within an unlabeled language corpus, these models aim to expose the underlying relationships and contextual patterns associated with words. These embeddings facilitated the encoding of contextual information, enabling NLP systems to discern subtle distinctions in meaning and context [24]. The integration of RNNs and word embeddings collectively marked a critical juncture in the enhancement of NLP capabilities, laying the groundwork for the following advancements in the field. Widely utilized in tasks such as sentiment analysis [25], [26], [27];

document classification [28] and machine translation [29], [30] word embeddings have become an essential tool in NLP.

The emergence of transformer architectures stands as a game-changing milestone in the progress of NLP. Bidirectional Encoder Representations from Transformers (BERT) [31] and Generative Pre-trained Transformers (GPT) [31], [32] are the influential models in the transformer architectures. These architectures have achieved groundbreaking success by adeptly capturing contextual information, thereby enhancing the understanding of the complexity of language. The transformative impact of these architectures is apparent in the transformers-based model applications across diverse tasks within NLP. The pre-trained language model approach, wherein models are initially trained on vast large datasets and subsequently fine-tuned for specific tasks, has emerged as a dominant method such as offensive language detection [33], text summarization [34] and named entity recognition [35]. The success of transformers can be attributed to their attention mechanism, allowing simultaneous consideration of all positions within a sequence and capture long-range dependencies more effectively than their predecessors. Beyond their innovative mechanisms, the main factor contributing to the success of these models lies in their training with gigantic datasets. This shift in architecture and training strategies underscores the efficacy and adaptability of transformer-based models, emerging as a state-of-the-art development in enhancing the capabilities of NLP systems.

In recent and ongoing developments, many remarkable trends have come to the forefront with the development of transformers in NLP. First, there is a notable expansion beyond textual data to encompass multimodal information, integrating images and audio into NLP frameworks. Models like CLIP [36] and DALL-E [37] serve as prime examples that integrate vision and language processing. Another critical aspect is the growing emphasis on developing explainable AI models within NLP.

Simultaneously, there is a persistent drive toward pushing the boundaries of model size and training data, resulting in increasingly large and powerful language models. This has resulted in models with extremely large numbers of parameters [6]. Ultimately, these models have started to be tailored for specific domains, demonstrating their capacity to comprehend and generate text relevant to these specialized fields.

1.1.3. Importance of NLP in Various Applications

NLP applications are utilized in various domains, enhancing the ability to understand and generate language to develop various processes and systems. NLP is fundamental in the development of virtual assistants and chatbots. It enables human-like interaction and conversation making these applications more intuitive and user-friendly. This not only improves user experience but also streamlines customer support and automates various tasks, contributing to increased operational efficiency for businesses [38], [39]. In addition to chatbots, NLP facilitates question-answering systems where the system understands natural language questions and provides relevant and accurate answers based on the available information [40], [41].

Moreover, NLP is extensively applied in search engines, where it is vital for interpreting user queries and providing search results that are both accurate and contextually relevant. Techniques such as Named Entity Recognition (NER) [35] and semantic analysis are key components in this process, significantly enhancing the precision and relevance of the search results.

The impact of NLP extends to machine translation [29], [30] information extraction, and text summarization [34], facilitating cross-language communication, extracting structured information from unstructured data and generating concise and coherent summaries to extract key information from longer texts. NLP is also employed for text classification to categorize and classify text into predefined categories. This is applied in spam detection [42], sentiment analysis]25], [26], [27] and topic categorization, among other tasks.

Furthermore, NLP techniques are instrumental in integrating different forms of communication. They facilitate the conversion of speech and images into written text and vice versa, expanding the accessibility and usability of various applications [43], [44]. Speech recognition systems, integral to voice assistants, transcription services, and voice commands, heavily depend on NLP algorithms to comprehend and process spoken language accurately [45]. In a similar vein, NLP extends its capabilities to image processing, facilitating the extraction of valuable insights from visual data and fostering a more comprehensive understanding of multimodal content. Additionally, NLP contributes to text-to-image functionalities, where written descriptions or textual data are translated into visual representations [36], [37]. This is particularly valuable in applications such as content generation, where textual prompts are transformed into corresponding images.

In conclusion, NLP has a huge impact on technology, influencing applications across diverse domains. From refining user interactions through virtual assistants and chatbots to enhancing search engines, NLP's impact is broad and transformative. Its applications also extend to machine translation, information extraction, text summarization, text classification, and speech recognition, showcasing its versatility and ongoing contributions to technological advancements.

1.2. Development of NLP Models in Named Entity Recognition

1.2.1. Definition of Named Entity Recognition (NER) and Focus on NER Applications

NLP offers valuable insights and efficiency enhancements across diverse domains in various applications as mentioned in section 1.1.3. One of the methods that has an important role in the domain of NLP is NER. NER is a core NLP task that focuses on recognizing and categorizing entities in a text, including People, Organizations, Locations, Dates, and others [46]. NER is an essential task in various NLP applications, including information retrieval, question-answering systems, sentiment analysis, and more. By identifying named entities in text, NER can help to extract useful information and insights from unstructured text data.

Consider a medical document that includes the following sentence "The patient has a family history of diabetes, and they are currently taking medication for hypertension." In this context, NER can be applied to automatically identify and categorize crucial entities related to medical information. For instance, NER can recognize "diabetes" as a medical condition and "hypertension" as another medical condition. Additionally, it can identify "medication" as a relevant concept, providing insights into the patient's current treatment. By automating the extraction of such entities, NER assists healthcare professionals in quickly comprehending key information from medical texts. This not only speeds up reviewing patient histories but also contributes to a more organized and structured representation of medical data, which is vital for clinical decision support systems and research endeavors in the healthcare domain.

Through the automated identification and categorization of named entities, NER is essential in building structured knowledge bases, thereby facilitating more informed decision-making processes. By identifying named entities in text, NER allows for the retrieval of valuable insights and information from unstructured data. The successful execution of NER has far-reaching implications for numerous applications.

In information retrieval and search engines, NER is crucial for enhancing search accuracy and relevance [47], [48]. Accurate entity identification and categorization allow search algorithms to offer more precise and contextually appropriate results. This is particularly beneficial in fields like journalism, research, and data mining, where efficient retrieval of information is paramount.

In the context of social media, news and sentiment analysis, NER contributes to a deeper understanding of user-generated content [49], [50]. Identifying entities in social media posts allows for the extraction of valuable insights about trends, public opinions, and sentiments related to specific entities. Businesses, marketers, and policymakers can leverage this information to make conscious decisions and respond effectively to public sentiment.

NER also plays a significant role in machine translation applications [51]. By accurately recognizing entities in a source language, translation models can better preserve the meaning and context when generating translations. This is particularly crucial for translating content in domains with specialized terminology, such as legal or medical texts.

NER is also crucial in the biomedical field. By accurately identifying and categorizing entities such as genes, proteins, diseases, and pharmaceuticals, NLP systems contribute significantly to biomedical information retrieval, aiding researchers, clinicians, and healthcare professionals in conducting comprehensive analyses, supporting diagnostics, and facilitating informed decision-making processes [52] [53].

In summary, NER serves as a cornerstone for deriving valuable information from raw text. The continued advancement of NLP techniques, particularly in NER, holds the promise of further optimizing and revolutionizing various industries and applications.

1.2.2. Notable NLP Architectures for NER

NER research has witnessed several key milestones that have shaped its evolution and contributed to its current state-of-the-art status. These milestones represent significant progress in the advancement of techniques and systems for identifying and classifying named entities within natural language text.

1.2.2.1.Rule-Based Approaches

As natural language processing techniques have progressed, corresponding advancements have been made in the area of NER. In the early stages of NER research, rule-based approaches dominated [46]. Systems were designed with predefined linguistic rules and heuristics for detection and classifying named entities such as ANNIE NER tagger [54]. These rules often relied on patterns in capitalization, syntactic structures, and contextual cues. Although limited by their rigidity, these systems laid the foundation for recognizing basic named entities such as names of people, organizations, and locations.

1.2.2.2.Transition to Statistical Models

As developments progress, there was a transition towards statistical models in the field of NER research. Probabilistic models, including Hidden Markov Models (HMMs) [3] and Maximum Entropy Models (MEMs) [4] gained popularity. These models utilized statistical patterns observed in large corpora to make predictions about the presence and category of named entities [55]. This transition allowed for greater adaptability and improved performance, as statistical models could learn from diverse data sources.

1.2.2.3.Rise of Machine Learning Algorithms

As machine learning algorithms advanced, their integration into NER research became prevalent. Supervised learning techniques, such as Support Vector Machines (SVMs) and Conditional Random Fields (CRFs) emerged as key contributors to the incorporation of machine learning algorithms into NER research. These approaches enabled the automatic learning of patterns and relationships from labeled training data, enabling NER systems to adapt and perform across diverse linguistic contexts and domains [56]. The shift to machine learning algorithms marked a transformative phase, enhancing the adaptability and performance of NER systems such as Stanford NER tagger [57] in extracting named entities from diverse and complex textual data.

The most significant milestone in NER research came with the widespread adoption of deep learning techniques. Deep neural networks, particularly recurrent and convolutional architectures, are noticeable innovations in the field by automatically learning hierarchical features from text. RNNs and LSTM networks improved the capture of contextual dependencies, leading to significant advancements in NER accuracy and high performance for the NER tagger systems such as SpaCy NER tagger [58].

These architectures are successful at capturing long-distance dependencies and sequential patterns, enabling NER models to distinguish relationships between words and phrases even if they are separated by significant distances. The emergence of deep learning has not only improved the performance of entity recognition models but also brought about a shift towards end-to-end learning, where models can extract patterns directly from unstructured text, reducing the reliance on comprehensive feature engineering [7]. The inherent capacity of deep learning architectures to automatically extract relevant features from raw data has greatly streamlined the development process, allowing for more efficient and adaptable NER solutions [7], [8], [9], [10], [11], [12].

Furthermore, the emergence of transformer-based models such as BERT has had a significant impact not only on NER but also on all tasks within NLP [31], [59], [60]. Attention mechanisms in these models enable the capture of comprehensive contextual information, facilitating the understanding of relationships across the entire sequence of text [31]. The contextual embeddings generated by transformers contribute significantly to the understanding of language, making them highly effective in the area of NER.

1.2.2.4. Pre-trained Language Models

Pre-trained language models have gained significant focus recently, which are trained on substantial and diverse text sources before being fine-tuned for targeted NER tasks. This approach has consistently proven effective in understanding complex linguistic structures and achieving state-of-the-art performance on a variety of NER benchmarks. The transition to pre-trained language models represents a more data-driven and context-based technique than traditional rule-based or supervised learning approaches. The emergence of pre-trained Transformers-based models has reshaped NER research and made significant progress toward more accurate and context-aware entity recognition.

These developments collectively reflect the dynamic change of NER research, showcasing the continuous integration of innovative approaches and technologies to increase the accuracy and versatility of NER systems.

1.2.3. Comparative Analysis of NER Models

In the field of NER, diverse methodologies have been employed, each bringing distinct strengths and limitations to the forefront.

Rule-based approaches stand out for their simplicity and transparency. Their straightforward design makes them easy to implement, and the explicit criteria defined by rules offer transparency in decision-making [61]. Chiticariu et al. [62] demonstrated that complex NER annotator can be successfully adapted for various domains, surpassing the quality of existing benchmarks. Unlike more complex models, rule-based systems operate on explicit, predefined rules, simplifying the process for users to comprehend the reasoning behind the system's output. This can be advantageous for gaining trust, facilitating auditing, and ensuring accountability in various domains. However, the creation of rules for this annotator remains a manual and time-consuming process. This challenge arises from their strict adherence to predefined rules, which might not encompass the linguistic diversity present in diverse contexts. Additionally, the sustained maintenance of these systems becomes an ongoing challenge, demanding frequent updates to accommodate evolving language structures [62]. The static nature of rule-based systems makes them less flexible in dealing with the dynamic and evolving nature of language.

Statistical models such as HMMs and MEMs capture statistical patterns by utilizing comprehensive corpora. This adaptability allows them to learn from data, providing greater flexibility than rule-based systems. However, they may have difficulty capturing long-term dependencies and subtle contextual information due to their local information focus [56]. These models rely on conditional probabilities and fixed window sizes, limiting their ability to consider distant and broader contextual dependencies in sequential data like language. Moreover, the success of these statistical models depends on the quality of the training data and suboptimal training data can adversely impact their performance. Additionally, these models have difficulty accommodating variable length sequences. This limitation hinders the models' ability to accurately model complex linguistic relationships within extended text spans [56].

Machine Learning (ML) algorithms offer enhanced adaptability by generalizing patterns from labeled data. They outperform rule-based and statistical approaches due to their data adaptation and generalization. CRFs, which are widely used ML techniques in NER tasks, outperform HMMs by considering richer contextual information. Unlike HMMs, which assume independence between observations, CRFs capture relationships within sequential data by bringing out dependencies among sequential data. This ability to account for dependencies between neighboring data points allows CRFs to better handle the complexities of natural language [63], [56]. With these advantages, CRFs become a superior choice in NER tasks compared to statistical models and they contribute to enhanced accuracy and performance in information extraction from unstructured text. Additionally, numerous NER models based on deep learning incorporate a CRF layer for tag decoding, often positioned above a bidirectional LSTM layer [12].

In addition to these advantages of ML models, they also have some drawbacks compared to deep learning-based models. One notable limitation is their reliance on handcrafted features, requiring domain expertise and substantial effort in feature engineering. Deep learning models, especially in the context of NER, are rooted in their capacity to self-extract significant features from input data. Unlike traditional ML models deep learning techniques, such as neural networks, can autonomously identify and extract hierarchical and abstract representations directly from the raw input. In this way, deep learning models can recognize complex patterns, relationships, and contextual dependencies within the text without explicit feature engineering [64]. This capacity for automatic feature learning is particularly powerful in tasks where the relevant features might be complex or challenging to articulate manually. Moreover, ML models may encounter challenges when dealing with large-scale datasets. In contrast, deep learning models are more efficient in handling extensive datasets, facilitating more effective training and generalization [64].

On the other hand, while deep learning models can learn the feature representations automatically, their black-box nature can make them less interpretable compared to traditional ML models. The requirement for significant computational resources during the training process, particularly for deep neural networks, can also be a significant drawback. Furthermore, to attain optimal performance, deep learning models often require extensive labeled data [64]. However, achieving the labeled data in domains with limited annotated datasets can be challenging [65]. Moreover, their high computational intensity and computational resources are one of the notable considerations.

The latest advancements in NLP have highlighted the remarkable efficacy of pre-trained language models, making them significantly more effective in the domain of NER. Pre-trained language models are typically trained on large-scale textual datasets before being fine-tuned for specific tasks [31], [12]. This pre-training process enables them to capture extensive linguistic patterns and contextual information. Training these models with gigantic datasets facilitates better transfer learning for the NLP tasks.

While both traditional ML and deep learning approaches often demand extensive annotated data for effective training, pre-trained language models can leverage their pre-training on vast corpora to grasp general language structures. This reduces the dependence on extensive annotated datasets and allows for effective performance even in domains with limited annotated data. Since pre-trained language models can be fine-tuned for specific tasks with relatively smaller labeled datasets. Through this fine-tuning, the model is adapted to address the specific challenges of the target task, which enhances its performance in identifying named entities in defined contexts [66].

Nevertheless, it is essential to note that pre-trained language models are not without their challenges. The computing power needed for training and fine-tuning can be substantial, and their large model sizes may cause deployment challenges in resource-constrained environments [67]. Additionally, fine-tuning on smaller datasets may lead to overfitting, emphasizing the need for careful consideration and validation during the fine-tuning process [68]. Despite these considerations, the overall advantages of pre-trained language models in enhancing NER performance make them a promising and impactful tool in NLP tasks.

1.3. Rationale for the Need of NER in E-commerce Queries

1.3.1. Challenges in E-commerce Search Queries

The development of e-commerce has led to a rapid increase in the number of online transactions, making search engines an essential tool for consumers to find products and services. The application of Named Entity Recognition (NER) on user queries becomes particularly significant as e-commerce platforms are heavily dependent on user search queries for product discovery and enhancing user experience. These queries are often short, concise, and expressed in natural, unstructured language, making accurate entity identification and classification crucial for improving search relevance and user satisfaction. However, the complexity of user input introduces challenges, such as ambiguity where a single query can refer to multiple entities or categories. For example, a search for "apple" might be seeking information on the fruit, the technology company, or a specific electronic device. Additionally, variability in user input, including synonyms, abbreviations, or alternative terms for the same entity, requires NER models to be robust enough to capture diverse linguistic expressions.

Additionally, E-commerce platforms offer an extensive array of products, each with unique characteristics, specifications, and brand variations. Identifying and categorizing named entities accurately within this diverse product landscape requires NER models to recognize a wide range of entities related to products, brands, and specifications, adding complexity to the task. Also, E-commerce platforms frequently update their product catalogs, introduce new items, and modify existing listings. This dynamic nature requires NER models to adapt to changes and identify entities within an unstable landscape effectively.

Moreover, E-commerce search queries are generally concise due to user preferences and the nature of search engine interfaces. The brevity of queries can lead to a lack of context, making it difficult to accurately identify and classify named entities. NER models should be successful in contextual understanding even when presented with limited information.

Overcoming these challenges requires the development of advanced NER models that leverage contextual embeddings, pre-trained language models, and domain-specific knowledge. Additionally, being able to handle short and contextually ambiguous queries is crucial to improving the accuracy and relevance of named entity inference of e-commerce queries.

1.3.2. Role of NER in Enhancing Search Query Processing

In the context of e-commerce search queries, NER methods can contribute significantly to enhancing the quality of search results. By identifying and classifying the entities in search queries, search engines are better equipped to interpret user intent and produce results that are both more accurate and contextually relevant. By identifying the entities related to a product search, such as brand, color, size, and price, search engines can provide more precise results that match the user's preferences. Additionally, NER methods can be employed to examine search queries and identify trends in user behavior, such as the most popular products or brands, which can be valuable for businesses in developing their marketing strategies. Therefore, the application of NER methods in e-commerce search queries can significantly improve the user search experience and provide valuable insights for businesses.

1.3.3. Significance of NER in Turkish Language Processing

There are various techniques and algorithms for performing NER, including rule-based systems, statistical models, and deep learning models in different usage areas for the Turkish language. Güneş and Tantuğ [69] proposed a named entity recognition model for the Turkish language, which utilized artificial neural networks with word embeddings to detect entities such as Person, Location, and Organization known as ENAMEX types. The study involved experimenting with various LSTM variations, ultimately achieving a high F1 score of 93.69% using the deep Bi-directional LSTM model. To develop the NER model, the author utilized labeled datasets created by Tür, et al. [70] and Seker and Eryiğit [71]. Akkaya and Can [72] proposed a transfer learning methodology to address the challenge of recognizing rarely seen entities in a dataset. The authors avoided using language-based features, which improved the performance of the NER system when applied to languages with complex morphology, such as Turkish. The NER model's architecture was based on the LSTM, with the cooperation of an additional CRF layer that provides transfer learning to the base model for the rare entities. To evaluate their proposed method, the authors used a Twitter dataset consisting of 55K tokens. The dataset was annotated with ENAMEX (Person, Organization, Location), TIMEX (dates and times), and NUMEX (numerical expressions). The methodology achieved an F1 score of 65.72%. Akdemir [73] introduced a joint learning method for dependency parsing and named entity recognition (NER), using separate datasets for each task and a shared encoder for feature extraction. By combining BiLSTM and CRF layers, the model incorporated syntactic information from dependency trees, improving NER performance and surpassing standalone NER models.

Çelikmasat et al. [74] concentrated on named entity extraction in the biomedical field, targeting entities like drugs and diseases. They enhanced a BiLSTM+CRF model by integrating transformer-based language models, including domain-specific BioBERT, and experimenting with various embeddings and architectures, such as FastText and Graph Convolutional Networks. The study demonstrated that incorporating pre-trained models, particularly BioBERT, significantly improved performance by 4% to 13% compared to the baseline BiLSTM+CRF model across multiple biomedical datasets. Dinç [75] investigated transformer-based deep learning models for detecting financial entities in various Turkish datasets. The study introduced two newly annotated datasets from financial news texts, using both BIO schema and raw labels. The agreement among annotators was assessed, and the models were evaluated with both annotation types to analyze how annotation format affects performance. Çarık [76] aimed to enhance transformer-based models' performance by incorporating external knowledge from Wikipedia in an unsupervised manner. Two methods were introduced: one for general entity selection from Wikipedia and another for emphasizing contextually relevant pages. These approaches significantly improved results on the MultiCoNER dataset compared to standard transformer models. The study also involved constructing a new Twitter dataset and comparing various transformer-based models with a BiLSTM-CRF architecture. The results showed that the BERT-CRF model outperformed other approaches, while the BERT-BiLSTM-CRF model lagged behind.

Topçu and El-Kahlout [77] developed TR-SEQ, a Turkish language dataset aimed at enhancing the performance of NER models. The authors conducted research on Turkish search engine queries to recognize a specific set of named entities. They collected search engine entries from Yaani, determined categories for search engine needs, and labeled the dataset using an in-house annotation method. The dataset consisted of 97,428 search engine queries, which were labeled by multiple individuals and verified through a verification study. The dataset was labeled using 7 named entity types, including organization, person, production art music and location which frequently occurred in the dataset. The authors used a BERT model based on transformers to achieve a high-performance NER tool, resulting in an accuracy of 90.41%. The BERTurk model was fine-tuned using their labeled dataset to train the Turkish NER system. They also experimented with the pre-trained ELECTRA [78] model to compare their dataset's performance on different deep architectures.

1.3.4. Existing NER Models in E-commerce Query

This section explores the current state of NER models applied to e-commerce queries, highlighting their strengths and limitations. Previous studies are examined in terms of their methodologies, datasets, and language considerations that have influenced the landscape of entity recognition in e-commerce contexts. Furthermore, this section addresses challenges encountered by existing models, including their adaptability to diverse linguistic structures, scalability, and domain specificity.

Bhange et al. [70] built an end-to-end framework to enhance search engine performance in e-commerce by applying NER to the search query. They prepared three sets of training data to have a large volume, high-quality labeled and high coverage of label values data and each dataset represents the features respectively. They constructed these datasets using homedepot.com. Two main entities which are Brand and Product type for e-commerce used to label the dataset. The three sets of training data were iteratively trained with a bidirectional GRU-CRF model to deliver a deep learning model for production. By deploying the most effective model as a real-time web service for homedepot.com, the F1 score was enhanced from 69.5% to 93.3% on the holdout test data. Zhang et al [80] proposed a NER model for e-commerce specifically electronic product descriptions. In this study, the authors employed an iterative bootstrapped positive-unlabeled learning algorithm that incorporates domain-specific linguistic features to efficiently expand the seed dictionary. The initial seed dictionary for NER was constructed using domain-specific knowledge and manually annotated data. The algorithm then iteratively added the most confidently classified samples from the unlabeled set to the positive set and retrained the model using the expanded positive set. This bootstrapping process continued until the desired performance was achieved. The NER task focuses on identifying four named entities: product, component, brand, and attribute. The authors utilized BERT to encode the input text and subsequently classified tokenized inputs using a Long Short-Term Memory (LSTM) network. In contrast to utilizing the Conditional Random Field (CRF) layer to obtain entities, the authors treated this task as a token-level classification. The implemented model achieved an average F1 score of 72.02% on the enlarged dataset, which demonstrated a 3.63% improvement compared to a baseline BiLSTM classifier model. Furthermore, the model showed an average increase of 4.96% in recall. Wen et. al. [81] proposed a NER model for e-commerce that aims to identify product item entities, such as brand, color, size, and texture, to understand customer intent from user queries. The authors suggested an end-to-end machine learning system that generates labeled large-scale data without the need for human labeling, thereby improving search relevance and conversion. The authors acknowledged the unstructured and noisy characteristics of query data, as well as the challenge of working with short text that lacks context and has low coverage of item aspects. To enhance the accuracy of tagging within the dataset, the authors employed FastText to classify the user's intent. To predict item aspects, a Bi-LSTM-CRF model was used, achieving an accuracy of 99.5%.

In the domain of e-commerce, Siddiqa et al. [82] proposed an effective NER solution that leverages character and subword embedding methods. To embed the inputs, they utilized character-level CNN, FastText, and Byte-Pair Encoding (BPE). NER system of this study is based on self-training to handle of out-of-vocabulary (OOV) and noisy labels problems. Specifically, this study aims to overcome label scarcity in NER datasets by implementing a teacher-student self-training framework. This approach employs a small, annotated dataset to label a much larger dataset. Training of the teacher model is conducted with a small, labeled dataset. The model assigns labels to the unlabeled data, named as pseudo-labels. Training of the student model is employed with a combination of labeled and pseudo-labeled datasets until convergence is achieved. The self-training models are based on the BiLSTM with the best performance embeddings. The authors conducted experiments on three different datasets and achieved significant improvements in the weighted F1 scores, namely 15.50%, 3.42%, and 8.48% compared to baselines.

Overall, the advancement of NER techniques in the area of e-commerce aims to enhance search engine performance by accurately identifying named entities such as products, brands, and attributes in user queries. To achieve this, various techniques have been employed such as deep learning models based on transformers, bootstrapped positive-unlabeled learning algorithms, and self-training frameworks. In addition to developing NER models, the creation of high-quality labeled datasets is crucial for improving NER performance.

1.4. Aim of the Thesis

1.4.1. Objectives of the Research and Contribution to the Field of NLP and E-commerce

This thesis focuses on developing and evaluating a NER system for e-commerce Turkish search queries. By harnessing advanced NLP techniques and ML algorithms, the proposed NER system seeks to identify and categorize specific entities within the complex structure of Turkish search queries. The focus of the NER system will extend beyond generic entities and will emphasize the recognition of named entities crucial to the e-commerce domain, including but not limited to products, brands, sites, sizes, fabrics, prices, and product codes. The evaluation will focus on assessing the system's precision, recall, and F1 score, ensuring its effectiveness in accurately extracting and categorizing entities from a diverse range of user queries.

Additionally, another purpose aim of this study is to develop a labeled dataset in Turkish to support the development and evaluation of the proposed NER system. This annotated dataset will serve as an important resource for training and fine-tuning the ML models involved in the NER system.

The utilization of state-of-the-art NLP models, particularly pre-trained transformers-based models, to extract entities from search queries constitutes another main objective of this thesis. Leveraging pre-trained models offers a significant advantage by capitalizing on the linguistic knowledge encoded during their pre-training phases on large corpora. This approach empowers the NER system to grasp contextual and linguistic features specific to e-commerce Turkish search queries without requiring extensive training data specific to domain. By fine-tuning these pre-trained models, the thesis aims to utilize the contextual understanding established in the transformer models to achieve high performance for entity recognition.

Furthermore, most of the NER solutions have been developed for identifying entities such as persons, locations, and organizations in raw text, but these methods often fall short in the e-commerce domain. E-commerce search queries often involve a diverse range of entities specific to product-related information, including brand names, product categories, specifications, and other related details. This study also addresses the lack of research in this field for the Turkish language, as well as the lack of a similar dataset for Turkish.

This thesis aims to apply named entity recognition (NER) to e-commerce search queries in the Turkish language. To the best of our knowledge, no comparable studies have been conducted in this field specifically for the Turkish language. Although NER studies for e-commerce search queries are available in various languages, their scope is often limited. While this study addresses the lack of research in this domain for the Turkish language, it also underscores the absence of a similar dataset for Turkish. Therefore, this thesis fills a significant research gap by proposing an NER system designed for Turkish e-commerce search queries.

This thesis makes the following contributions:

1. Development of a specialized Named Entity Recognition (NER) system for e-commerce search queries in the Turkish language.
2. Introduction of an annotated dataset, providing significant resource for training and evaluating the proposed NER system.
3. Exploration and application of pre-trained word embedding and state-of-the-art transformers-based models to enhance the accuracy and adaptability of the NER system.
4. Addressing the scarcity of research in the specific domain of Turkish e-commerce search queries fills a notable gap in the existing literature.
5. Recognition of the unique linguistic challenges the Turkish language poses and the distinct nature of entities within e-commerce queries contribute insights for future research in this domain.

Overall, the findings of this study have substantial impacts on the e-commerce industry, demonstrating the potential advantages of utilizing NER techniques to improve the search experience for users and enhance business performance. By offering insights into consumer behavior and preferences, NER can help businesses optimize their product offerings and marketing strategies, while improving the ease and efficiency with which users can find their desired products.

2. MATERIALS AND METHODS

2.1. Dataset

2.1.1. Dataset Collection and Pre-processing

In this research, we employed internal search queries from sefamerve.com, a leading e-commerce retailer in Turkey specializing in modest clothing. We collected search queries corresponding to each product category, resulting in the compilation of an exhaustive dataset that encompasses a wide array of categories. To generate the dataset, we gathered queries searched on sefamerve.com within three months.

We removed duplicate queries, converted all data to strings, and normalized queries to lowercase. Due to users preferring to use English keyboards or typing Turkish characters without dots, the same query words can be written differently. For example, it is common to see "şal" (shawl) and "başörtü" (headscarf) searched as "sal" and "basortu" respectively. Such examples are frequently encountered in the dataset. In order to treat these query words as identical Turkish characters (ç, ğ, ı, ö, ş, ü) were transcribed to their corresponding ASCII representations.

2.1.2. Annotation Process

The queries were annotated into 17 distinct classes, each class representing a critical aspect of the search, including *size, height, gender, condition, price, discount, category, fabric, brand, seasonal, color, website, length, age, time, feature,* and *product_code.* These annotations were designed by identifying all the potential classes that might emerge within a query.

The dataset has been labeled utilizing the BIO tagging format. Examples for BIO tagging format are illustrated in Table 2.1. The BIO tagging format, commonly used in NLP tasks like named entity recognition (NER), is a systematic method for annotating and labeling words or tokens in a text corpus to signify their roles about named entities.

Table 2.1. BIO tagging format example.

BIO tagging on example queries			
Query (Turkish)	çiçek	desenli	başörtü
Query English	floral	patterned	headscarf
BIO tags	B-feature	I-feature	B-category
Tags	Feature		Category
Query (Turkish)	medine	ipeği	ferace
Query English	medine	silk	abaya
BIO tags	B-fabric	I-fabric	B-category
Tags	Fabric		Category
Query (Turkish)	kadife	tunik	modelleri
Query English	velvet	tunic	models
BIO tags	B-fabric	B-category	O
Tags	Fabric	Category	Outside

BIO stands for "Beginning, Inside, Outside." In this format, "B" (Beginning) marks the first word of a named entity, signifying the entity's initiation. "I" (Inside) labels words within the named entity, excluding the first word, indicating their participation in the entity. "O" (Outside) represents the non-entity terms highlighting that they exist outside of any entity. This format annotates the query “çiçek desenli başörtü” (floral patterned headscarf) with "B-Feature" for "floral," "I-Feature" for "patterned," and "B-Category" for "headscarf," aiding in the structured recognition and classification of entities.

2.1.3. Dataset Statistics

The dataset contains 10,435 samples with 4,821 unique words and a maximum query length of 21 tokens. Figure 2.1 presents a graphical representation of label frequencies within the training dataset, presented as percentages. As observed in the figure, labels with a higher percentage are predominantly associated with the 'B' (beginning) of the tagging sequence. Specifically, labels for 'category,' 'feature,' and ‘color’ (outside) emerge as the most prevalent categories within the queries, underscoring their significance. In contrast, 'I' (inside) tags exhibit comparatively lower frequencies within the dataset.

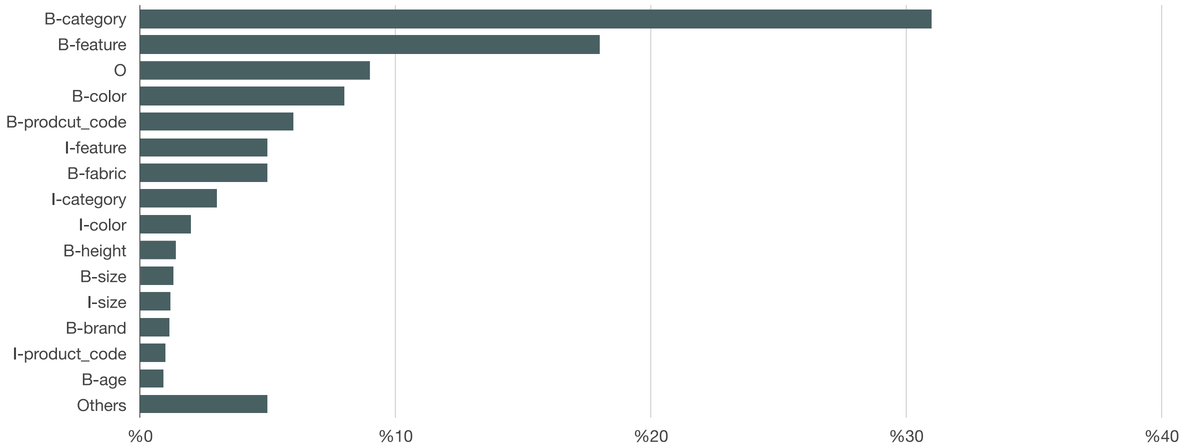

Figure 2.1. Label frequencies of the BIO formatted dataset as percentages.

The dataset was randomized by shuffling the data to obtain an unbiased distribution of queries across the subsets. For model development and evaluation, the dataset was separated into three parts: 60% was allocated for training, 20% for validation, and 20% for testing. Table 2.2 shows the statistics of the dataset. The frequencies of entities are shown in this table for each train and test set. The distribution of entities is evenly balanced between the training and test sets.

Table 2.2. Dataset statistics.

Tags	**Train (Frequency)**	**Test (Frequency)**
Category	5,692	1,962
Feature	3,977	1,401
Color	1,627	518
Product code	1,166	361
Fabric	973	274
Size	432	146
Height	261	94
Brand	210	74
Price	202	65
Discount	193	57
Age	169	50
Seasonal	122	47
Gender	108	42
Time	66	24
Website	49	13
Condition	18	6
Queries	6,261	2,087
Tokens	16,763	5,558
Named Entities (NE)	15,265	5,134
Unique NE	4,021	1,833

2.2. Model Architecture

This section presents an in-depth explanation of the model architectures employed in our study. Figure 2.2 illustrates the comprehensive overview of our proposed method, covering all stages from data preparation to the implementation of the models. Firstly, the dataset is prepared and separated into training and test sets. The training dataset is used to train the proposed NER models. We proposed four different main NER prediction models. For the baseline, we used BiLSTM and BiLSTM with CRF. Also, we combined pre-trained word embeddings with the baseline models. Thirdly, we utilized various state of art transformers models. Finally, we proposed a hybrid model which combines BERT with BiLSTM+CRF. Entity prediction of each model was evaluated with the test set.

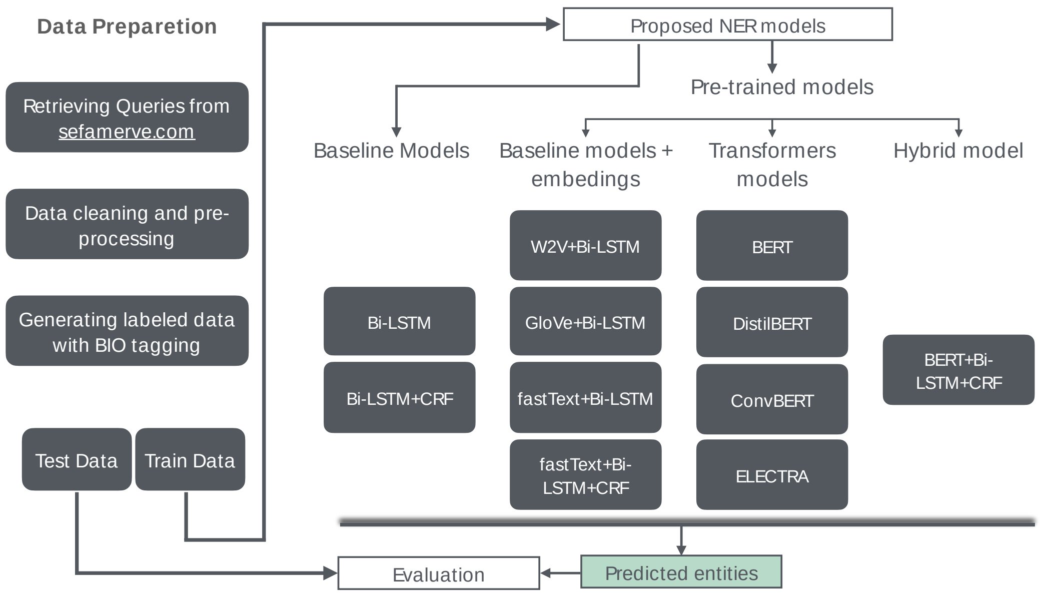

Figure 2.2. Overview of the proposed method.

2.2.1. Baseline Models

In this section, the baseline models were introduced for the NER experiments for e-commerce search queries in Turkish. These models serve as foundational benchmarks against which the more advanced architectures are evaluated.

2.2.1.1.Bi-LSTM

The Bi-LSTM model is a specialized recurrent neural network architecture. It uniquely captures sequential dependencies by processing data in both forward and backward directions [83]. The bi-LSTM model comprises multiple LSTM layers processing the input sequences in both directions. We utilize this model to establish a baseline understanding of how well a traditional sequence labeling approach performs on our Turkish e-commerce search query dataset.

In RNNs, the output is shaped by the current input as well as the hidden states from earlier time steps. During time t, the hidden layer combines the results from the previous step $t-1$ with the input at time t. Consequently, the input at time $t-1$ impacts the decisions made at time t. To put it differently, the outputs result from a combination of current and past information, ensuring the retention of sequential information in these networks [14]. The LSTM is developed to address the vanishing gradient issue of traditional RNNs. LSTMs use memory cells with gating mechanisms to selectively store, read, and write information. The key components of an LSTM unit include the input gate (i_t), forget gate (f_t), cell state (C_t), output gate (o_t), and hidden state (h_t). The formulas governing the LSTM computations at each time step are as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (2.1)$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (2.2)$$

$$\tilde{C}_t = \sigma(W_c x_t + U_c h_{t-1} + b_c) \quad (2.3)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.4)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \quad (2.5)$$

$$h_t = o_t \odot \tanh\,(C_t) \quad (2.6)$$

Here, x_t is the input at time step t, h_{t-1} is the hidden state from the previous time step, $W,\ U$ and b are weight matrices and bias vectors which are learned during the training, and σ is the sigmoid activation function. The tanh function is denoted by $tanh$. The operator $\odot$ implies the element-wise product. Figure 2.3 illustrates the LSTM recurrent unit.

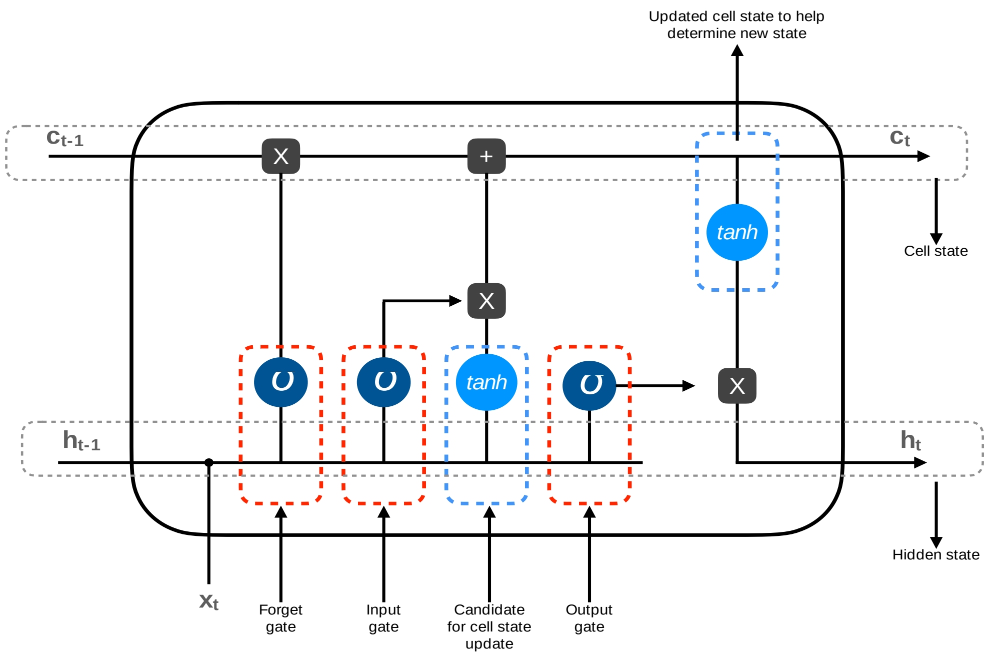

Figure 2.3. LSTM recurrent unit.

For the Bi-LSTM, the forward and backward computations are as follows:

$$h_t^f = LSTM\_forward(x_t, h_{t-1}^f) \tag{2.7}$$

$$h_t^b = LSTM\_backward(x_t, h_{t+1}^b) \tag{2.8}$$

The final hidden state h_t is obtained by concatenating the forward (h_t^f) and backward (h_t^b) hidden states at each time step. By processing input bidirectionally, the model gains a deeper understanding of the sequence, which enhances its entity recognition capabilities.

2.2.1.2.Bi-LSTM+CRF

To integrate the CRF layer into the existing model architecture, we extend the Bi-LSTM model. The motivation behind incorporating a The CRF layer's strength is in modeling dependencies between sequential labels in a more coherent manner than traditional softmax layers. It models the probability of a label sequence conditioned on the given input sequence. The Bi-LSTM model, with its forward and backward information flow, is highly effective at understanding context from both preceding and following tokens in a sequence. However, it may struggle to optimize global label dependencies, as the softmax layer independently assigns labels to each token. This limitation can result in suboptimal predictions, especially when the correct labeling involves complex interactions between multiple tokens. Figure 2.4 shows the architecture of the Bi-LSTM+CRF model.

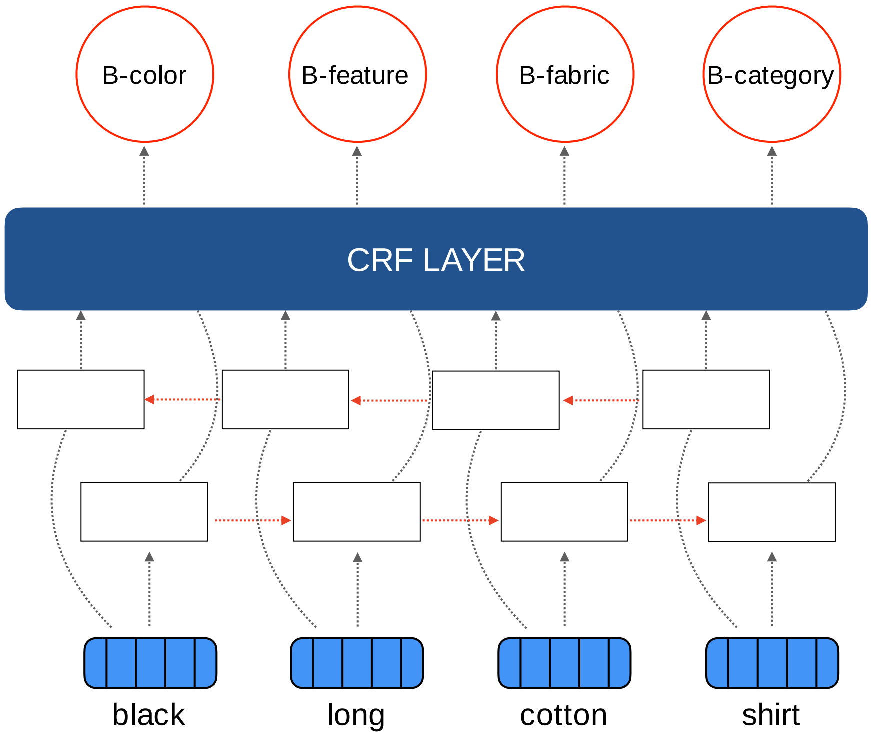

Figure 2.4. Bi-LSTM+CRF model architecture.

The CRF layer adds a pairwise scoring mechanism that evaluates the compatibility between neighboring labels. This scoring considers both the emission scores from the Bi-LSTM layer (indicating the suitability of each label for a given token) and the transition scores (indicating the likelihood of transitioning between specific label pairs). The overall objective is to find the label sequence that maximizes the global score, ensuring coherence in the final predictions.

Given a sequence of labels $y = (y_1,\ y_2, \dots, y_t)$ where y_t represents the label assigned to the token at time step t, the CRF model calculates the likelihood of the label sequence given the input sequence x is expressed as follow:

$$CRF\_score(x, y) = \sum_{i=0}^{t} A_{y_i, y_{i+1}} + \sum_{i=1}^{t} P_{i, y_i} \tag{2.9}$$

where A_{ij} represents the transition probabilities between label i to label j. P indicates the emission scores for each label at each time step provided by the output of the Bi-LSTM layer. The likelihood of the sequence y is determined by:

$$p(x|y) = \frac{e^{CRF\_score(x,y)}}{\sum_{y' \in y} CRF\_score(x,y')} \tag{2.10}$$

While each Bi-LSTM instance (time step) generates an associated output feature map along with CRF transition and emission values, the subsequent task involves decoding these individual time step outputs into a sequence of potential tags. To find the sequence of labels with the highest likelihood in a CRF model, the Viterbi algorithm was employed for this decoding process, to determine the final score associated with the chosen path through the possible tags. The algorithm iteratively computes the highest probability of reaching each label at every time step, considering all possible paths leading to that label. The recurrence relation for the Viterbi algorithm is as follows:

$$y^* = \arg\max\left(CRF\_score(x, y')\right) \tag{2.11}$$

The final output of the Viterbi algorithm is the sequence of labels that maximizes the joint probability $p(x|y)$. This optimal label sequence represents the predicted named entities for the input sequence.

2.2.1.3.Word Embeddings

Previous models do not have pre-trained word embeddings, the embeddings were randomly initialized. On the Bi-LSTM models, a layer of pre-trained word embeddings was added to observe the contribution of the learned word representations. We utilized three different publicly available embedding models trained with the Turkish language dataset. The embedding models that were used in this study are the following:

- Word2Vec [21]: The model developed by Google, converts words into high-dimensional vectors within a continuous space, clustering words with similar meanings closely. This model is trained on large corpora of text using neural networks, specifically either the Continuous Bag of Words (CBOW) or Skip-gram architecture.
- GloVe [23]: It stands for Global Vectors for Word Representation, is designed to represent words as vectors by leveraging global co-occurrence statistics from large-scale text data. Unlike Word2Vec, which learns word embeddings based on local context, GloVe considers the entire corpus to generate word vectors. This method emphasizes the relationships between words based on their co-occurrence probabilities in the corpus.
- FastText [84]: It generates word vectors by considering the co-occurrence of words, maintaining both semantic and grammatical information. The model employs two

different approaches for computing vector representations: Skip-gram and Continuous Bag of Words (CBOW). During the training of word vectors, FastText utilizes character n-grams of different sizes, with n spanning from one to the full length of the word. This approach allows FastText to learn the vector representation of a previously unseen word by considering its n-gram combinations [84]. The inclusion of character n-grams in FastText makes it unique compared to other word embedding approaches like Word2Vec and GloVe..

2.2.2. Transformers-Based Models

2.2.2.1.BERT

To enhance our NER model's performance, we adopted Transformer-based models, specifically BERTurk, which is a Turkish-language variant of the BERT [31] model. The attention mechanism in BERT enables the model to assess the significance of various words in a sequence when processing each word. While traditional models process text sequentially in one direction, BERT simultaneously considers contexts from both directions. This bidirectional attention allows the model to grasp the context of each word by considering the entire input sequence. BERTurk has proven to be highly effective in capturing contextual information about the Turkish language [85]. This capability is provided by the attention mechanism during word representation, making BERT highly effective for various NLP tasks. BERTurk [86] has been pre-trained on vast amounts of Turkish text data, encompassing a substantial 35 gigabytes of text and comprising around 4,4 billion tokens. We fine-tuned BERTurk for our task. This fine-tuning process ensures that the The NER model acquires the capability to identify domain-specific named entities relevant to this thesis. further enhancing its precision and recall.

Moreover, we incorporate the Distilled version of BERT. DistilBERT is a distilled version of BERT designed to be more lightweight and faster while still maintaining good performance across different natural language understanding tasks [87]. In the NER experiments, we utilized DistilBERT to achieve a computationally efficient yet powerful alternative.

2.2.2.2.ELECTRA

ELECTRA [78] is an innovative Transformers-based model that deviates from traditional masked language modeling used in BERT architectures. ELECTRA comprises two main elements: the generator and the discriminator. In contrast to BERT, which estimates the original state of masked terms, ELECTRA employs the discriminator to determine if each term in the input, with some terms altered by a generator, has been modified. This modification extends to all input terms rather than just concentrating on the subset that is masked, making the task more efficient. The model was trained same corpus with BERTurk.

2.2.2.3.ConvBERT

ConvBERT [88] is a model proposed to address the limitations of BERT, particularly its heavy reliance on global self-attention mechanisms, which result in large memory usage and computational costs. To mitigate this, it introduced a novel span-based dynamic convolution mechanism for modeling local dependencies, replacing some of the self-attention heads. The model was trained same corpus with BERTurk.

2.2.2.4.BERT+BiLSTM+CRF

To increase the performance of pre-trained Transformer-based models, we proposed a corporation of LSTM, BERT, and CRF. By integrating features from both recurrent neural networks and transformer models, this architecture leverages their combined strengths with the structured predictions offered by the CRF layer. This model aims to capture both local and global contextual information for improved named entity recognition in Turkish e-commerce search queries.

In this architecture, the BERT component contributes by providing deep contextual embeddings for each token. The BiLSTM component serves as a sequential model that processes the input sequence and provides the emission scores for CRF. The CRF layer is then integrated to handle the structured predictions of named entities. The transition probabilities in CRF capture the dependencies between adjacent labels, refining the model's ability to recognize entity boundaries. Figure 2.5 shows the model architecture as an illustration.

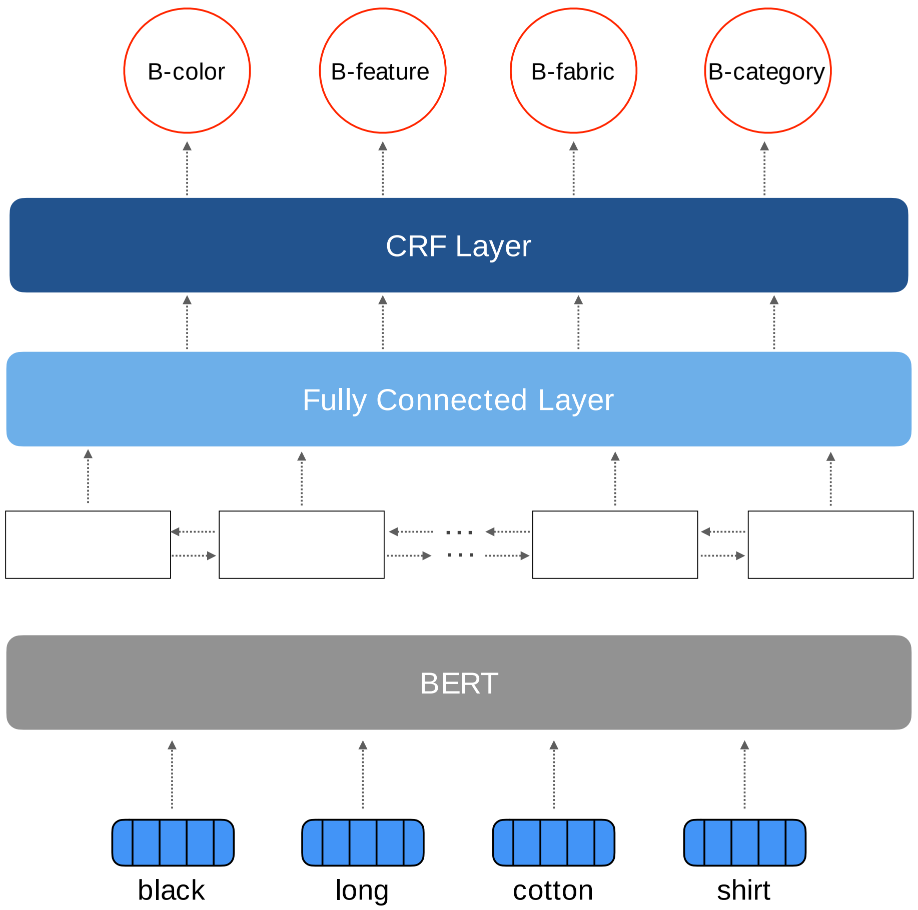

Figure 2.5. BERT+BiLSTM+CRF model architecture illustration.

2.3. Experimental Setup

2.3.1. Vectorization of Text Data

For the baseline models, the initial step involved the segmentation of search queries into individual tokens using whitespace as the delimiter. Subsequently, each token was assigned a unique index number to facilitate the models' understanding of the input. To ensure uniformity in the input dimensions for these models, the length of the search queries was fixed to the same length by employing zero-padding techniques.

Each token was mapped to its corresponding embedding to convert discrete textual units into continuous vector representations. With this embedding process, the models understand the inherent semantic relationships between tokens, laying the groundwork for feature extraction and pattern recognition during model training. Baseline models without pre-trained word embedding utilized randomly initialized word embeddings. This choice was made to explore the baseline performance without the influence of pre-trained embeddings.

For the transformers-based model, the WordPiece tokenizer [89], which is based on the subword tokenizer approach, was adopted for processing the input texts. The main idea behind the subword-based tokenization algorithms is to break down words into smaller subword units known as 'tokens'. Notable subword-based tokenization algorithms include WordPiece, Byte-Pair Encoding (BPE) [90], Unigram [91] and SentencePiece [92].

The WordPiece tokenization algorithm shares similarities with BPE, but there are key differences. BPE relies on the frequency of subword units. It starts with a vocabulary of individual characters and iteratively merges the most frequent pairs of adjacent subword units. This process is repeated until a predefined vocabulary size is reached. BPE essentially uses frequency as a guiding principle for merging subword units.

On the other hand, WordPiece incorporates probability or likelihood into the merging decision instead of relying on frequency. This approach allows WordPiece to make more informed decisions about which subword units to merge, potentially leading to a more contextually relevant and adaptive tokenization. The WordPiece tokenizer is used in NLP particularly state of art models such as BERT, DistilBERT and ELECTRA.

2.3.2. Hyperparameter Tuning and Training of the Models

2.3.2.1. Baseline Models

For the Bi-LSTM model, we employed a single layer of bi-directional RNN with a hidden size of 100 units in both forward and backward directions. We utilized word embeddings with a dimensionality of 100, initialized with random values, and did not employ pre-trained word embeddings. Zero-padding was utilized to ensure a fixed input length, and it was ignored throughout the model's training process. For the training process, we employed a batch size of 64 samples. Categorical cross-entropy was used as the loss function, and for the optimization the Adam [93] optimizer was utilized. The validation loss was monitored with a patience value of 5 for early stopping during the training. To process the data and implement the model, we utilized the Keras library.

In the Bi-LSTM+CRF model, we maintained the same architecture as the Bi-LSTM model in terms of the hidden layer size and batch size. We used Adam optimizer with $1\ x\ 10^{-3}$ learning rate and $1\ x\ 10^{-6}$ weight decay and the negative log-likelihood loss function was employed. The model was trained during 15 epochs. Implementation-wise, we utilized the PyTorch library.

2.3.2.2.Word Embeddings

Table 2.3. shows the training details of the word embedding models. Word2Vec[1] [94] was trained with the Wikipedia corpus and the CBOW method. Word2Vec [95] was trained using news articles, web pages, and books using the skip-gram model. GloVe[2] (Inzva) and fastText [96] were trained on Common Crawl[3] data which consists of web pages, books and Wikipedia corpus.

Table 2.3. Training parameters of the embedding models.

Models	Vocab Size	Dimension	Window Size	Negative Sampling	N-gram
W2V [94]	-	400	5	-	-
W2V [95]	2M	300	5	10	-
GloVe (Inzva)	570K	300	-	-	-
fastText [96]	2M	300	5	10	5

2.3.2.3.Transformers-Based Models

To fine-tune the BERTurk model for our NER task, we explored different settings. We experimented with various batch sizes (2, 8, 16, and 32), epoch numbers (5, 7, and 10), and optimizers (SGD and AdamW) with different learning rates $1\ x\ 10^{-3}$ and $1\ x\ 10^{-5}$. During training, the cross-entropy loss function was used. We obtained the best performance with SGD optimizer with $5\ x\ 10^{-3}$ and 7 epochs with an 8 batch size.

In adapting BERTurk for NER, we labeled only the first token in each input and assigned -100 labels to subtokens and special tokens like [CLS] and [SEP]. These -100 labels help us align BERTurk's tokenization with the NER task and they were ignored by cross entropy loss during the training. When evaluating the model's performance, tokens with -100 labels were simply excluded to ensure an accurate assessment.

In the DistilBERT model, the same hyperparameters delivered optimal performance for the BERT model were maintained.

---

[1] https://github.com/akoksal/Turkish-Word2Vec

[2] https://github.com/inzva/Turkish-GloVe

[3] https://commoncrawl.org/

In the ELECTRA model, the same experiments with the BERT model were conducted for hyperparameter tuning. The optimal configuration for the ELECTRA model entailed utilizing the AdamW [97] optimizer with a learning rate of $5\ x\ 10^{-5}$, a batch size of 8, and a training duration spanning 7 epochs.

In the BERT+Bi-LSTM+CRF model, the hidden dimension was configured as 128 x 2 for the Bi-LSTM. The model was trained for 9 epochs with a batch size of 16. AdamW optimizer was employed with a learning rate of $3\ x\ 10^{-5}$ and a weight decay of 0.01. We selected the model based on its optimal performance, which occurred during the 3rd epoch.

During training, we incorporated a linear scheduler designed to modulate the learning rate. This scheduler initiates with a learning rate equal to the optimizer's initial learning rate and linearly decreases it to 0 after an initial warm-up period. To achieve this, we specified a warm-up proportion of 0.01 and an Adam epsilon value of $1\ x\ 10^{-8}$ for the scheduler.

For all models, the training was conducted on a Google Colab GPU with a Tesla T4, which provided computational resources.

2.4. Evaluation Metrics

For evaluating the performance of the NER models; accuracy, precision, recall and f1-score metrics are used. Accuracy is the proportion of true predictions among the total number of samples evaluated.

$$Accuracy = \frac{true\ predictions}{total\ number\ of\ predictions} \tag{2.12}$$

Precision measures the accuracy of the model's positive predictions. It helps to measure the reliability of the model when it claims to have found entities. A higher precision signifies fewer false positives, indicating that the model's predictions are more reliable. Precision is determined by the ratio of true positive predictions to the total number of true positives plus false positives, expressed by the formula:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives} \tag{2.13}$$

Recall, also known as sensitivity or true positive rate, assesses the model's ability to capture all relevant instances of a particular entity in the dataset. In NER, recall is crucial for ensuring that the model doesn't miss identifying entities that are present. Recall is computed as the ratio of true positive predictions to the sum of true positives and false negatives, represented by the formula:

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives} \quad (2.14)$$

By taking the harmonic mean of precision and recall, the F1-score provides a comprehensive measure that reflects both false positives and false negatives. A higher F1-score indicates a model that performs well in both precision and recall. It is calculated as follows:

$$F1\_score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall} \quad (2.15)$$

In the experiments, both weighted and macro average F1-scores are used. Weighted F1-score is also calculated by computing the F1-score for each class, but it considers the class distribution in the dataset. The F1-score for each class is weighted based on the number of instances of that class, giving more importance to larger classes. The weighted F1-score is formulated as follows:

$$weighted\_F1\_score = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{N_i}{N}\ \ x\ F1\_score_i\right) \quad (2.16)$$

On the other hand, the macro avg. F1-score treats each class equally, making it suitable when all classes are considered equally important. It provides a balanced evaluation across all entity types. It is determined by calculating the F1-score for each class (entity type) individually and then averaging these scores across all classes. It gives equal weight to each class, regardless of the class distribution in the dataset. Macro avg. F1-score is calculated as follows:

$$macro\_average\_F1\_score = \frac{1}{N}\sum_{i=1}^{N} F1\_score_i \quad (2.17)$$

Since some entity types may be rare compared to others in NER tasks, the weighted F1-score helps ensure that the model's performance on less frequent entities is appropriately considered in the overall evaluation. On the other hand, the macro avg. F1-score provides an overview of the model's performance across all classes.

The model's performance was evaluated at the token-entity level, where assessments allowed for partial recognition of partially accurate predictions within the entity span. To illustrate, consider the search query “turkuaz yeşil keten tunik” (turquoise green linen tunic). If "turquoise" and "green" are correctly identified as describing the color, while "linen" and "tunic" are misclassified, the model would be credited partially based on the correctly identified tokens within the entity span.

3. RESULTS AND DISCUSSION

3.1. Experimental Results

In this section, we present the results of our experiments on NER for Turkish e-commerce search queries. The primary objective of this study was to evaluate the performance of various state-of-the-art NER models and architectures in accurately identifying named entities within the context of online shopping queries in Turkish. Through experimentation and evaluation, we aimed to demonstrate the effectiveness of different approaches and their suitability for addressing the challenges in this specific domain.

The results presented in this section offer insights into the performance of each model considered in our study, encompassing traditional approaches such as Bi-LSTM and Conditional Random Field (CRF), as well as cutting-edge transformer-based models like BERT, DistilBERT, and ELECTRA. Additionally, we explore the efficacy of hybrid architectures that integrate LSTM and BERT with CRF, leveraging both sequential and contextual information for improved NER performance.

3.1.1. Baseline Models

Table 3.1. shows the performance of the baseline models according to the different metrics. According to the results, the Bi-LSTM model achieved respectable precision, recall, and weighted F1-score, indicating its capability to correctly identify named entities. However, the macro avg. F1-score suggests that the model's performance varies across different classes, with room for improvement in capturing minority classes.

Table 3.1. Performance of the baseline models according to the accuracy, precision, recall, weighted f1-score, macro avg. f1-score.

Models	Accuracy	Precision	Recall	Weighted F1-Score	Macro avg. F1-Score
BiLSTM	0.8319	0. 8594	0. 8379	0. 8456	0. 7197
BiLSTM+CRF	**0.8743**	**0. 8897**	**0. 8603**	**0. 8721**	**0. 7727**

The Bi-LSTM+CRF model demonstrated significant improvements in all metrics compared to the Bi-LSTM model. The incorporation of the CRF layer enhanced the model's ability to capture sequential dependencies, resulting in better performance across all evaluation metrics. Overall, there is an approximate 4.36% average increase across all metrics by adding the CRF layer.

3.1.2. Baseline Models with Word Embeddings

Table 3.2. displays the performance of the different word embeddings. Based on the results of the BiLSTM with pre-trained embeddings, The FastText+BiLSTM model outperforms other BiLSTM with embedding models across various metrics. Surprisingly, the Word2Vec embeddings exhibits poor performance. The short and specific nature of e-commerce queries may be one of the reasons why models like fastText and GloVe outperform Word2Vec.

Table 3.2. Performance of the baseline models with different word embedding models according to the accuracy, precision, recall, weighted f1-score and macro avg. f1-score.

Models	**Accuracy**	**Precision**	**Recall**	**Weighted F1-Score**	**Macro avg. F1-Score**
W2V [94] + BiLSTM	0.7981	0.7981	0.7981	0.7964	0.7204
W2V [95] + BiLSTM	0.8299	0.8299	0.8300	0.8289	0.7742
GloVe (Inzva) + BiLSTM	0.8402	0.8433	0.8402	0.8407	0.8036
FastText [96] + BiLSTM	**0.8636**	**0.8622**	**0.8615**	**0.8605**	0.7878
FastText [96] + BiLSTM + CRF	**0.8765**	**0.8934**	**0.8653**	**0.8766**	**0.8060**

E-commerce queries typically come in the form of a list of product specifications, such as “çizgili uzun tunik” (long striped tunic). The meaning of such queries usually relies on the presence of the words they contain, but these words do not necessarily need a specific order. Therefore, models like FastText and GloVe can better handle such queries by learning the context between words even without considering word order.

The ability of fastText to handle out-of-vocabulary (OOV) words can be particularly effective in cases such as spelling errors and different spelling formats in queries. FastText constructs word vectors by utilizing sub-words of a word. Even in the absence of the word in the training data, fastText can predict its vector using the word's sub-words.

When the CRF model is added to the FastText+BiLSTM model, the resulting FastText+BiLSTM+CRF model emerges as the best-performing model among the base models. The positive impact of CRF, as seen in Table 3.1, is also evident here. The increase in macro avg. F1-score also indicates that the model obtains better performance across different and rare classes. Figure 3.1 shows the contribution of the CRF and word embedding models to the Bi-LSTM model. Adding a Conditional Random Field (CRF) layer to the Bi-LSTM model resulted in a 7.36% improvement in the macro F1-score, demonstrating the effectiveness of the CRF layer in enhancing sequence labeling performance. Furthermore, incorporating FastText embeddings alongside Bi-LSTM and CRF led to an even more significant increase of 4.54% in the macro F1-score, highlighting the combined power of contextual word embeddings and advanced sequence modeling techniques in improving model performance.

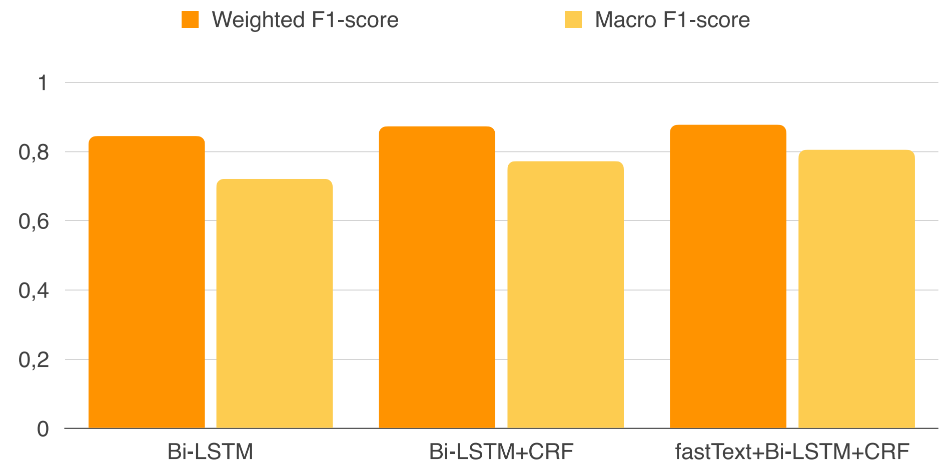

Figure 3.1. Weighted and macro avg. F1-scores of the Bi-LSTM-based models.

3.1.3. Computational Resources Comparison of the Baseline Models

Table 3. 3. shows the computational resource usage for the baseline models. According to the table, the PyTorch implementations offer remarkably faster training times compared to the Keras implementations. Despite being implemented on the CPU, PyTorch exhibits superior performance in training time. This suggests that for tasks requiring intensive computations, PyTorch could be the preferred framework, offering faster training.

Table 3.3. Computational resources comparison of the baseline models according to the RAM usage, average time for 1 epoch and GPU/CPU usage.

Models	Weighted F1-Score	RAM Usage (MiB)	Avg. Time (sec) (1 Epoch)	GPU/CPU
BiLSTM (Keras)	0. 8379	295	18	GPU
W2V [94] + BiLSTM (Keras)	0.7964	373	22	GPU
W2V [95] + BiLSTM (Keras)	0.8289	383	17	GPU
GloVe (Inzva) + BiLSTM (Keras)	0.8407	383	20	GPU
FastText [96] + BiLSTM (Keras)	0.8605	383	23	GPU
BiLSTM+CRF (PyTorch)	0.8603	512	3	CPU
FastText [96] + BiLSTM + CRF (PyTorch)	**0.8766**	600	5	CPU

Overall, FastText+Bi-LSTM+CRF has the highest weighted F1-score among all models, suggesting that leveraging FastText embeddings along with the CRF layer results in the most effective named entity recognition among the models.

3.1.4. Transformers-Based Models

Table 3.4 illustrates the performance of the transformers-based models. According to the results, all Transformers-based models outperform the baseline models in all metrics indicating the superior performance of the pre-training with huge amounts of data.

Table 3.4. Performance of the transformers-based models according to the accuracy, precision, recall, weighted f1-score, macro avg. f1-score and RAM usage.

Model	Accuracy	Precision	Recall	Weigthed f1-score	Macro avg. f1-score	RAM Usage (MiB)	Avg. Time (sec) (1 Epoch)
BERT	0.8874	0.9245	0.9182	0.9194	0.8412	2333	58
DistilBERT	0.8725	0.9151	0.9031	0.9075	0.7950	**1705**	**29**
ELECTRA	0.8892	0.9202	0.9228	0.9197	0.8427	3467	85
ConvBERT	**0.8897**	0.9187	0.9187	0.9187	0.7889	1709	68
BERT+Bi-LSTM+CRF	0.8991	**0.9256**	**0.9256**	**0.9249**	**0.8434**	6505	204

BERT+Bi-LSTM+CRF performs the best overall, with the highest accuracy, precision, recall, weighted F1-score, and macro avg. F1-score. Among the individual transformer models, ELECTRA has the highest precision, recall, weighted F1-score, and macro avg. F1-score. Figure 3.2 shows the performance of the transformers-based model. By integrating both BERT and a CRF layer into the Bi-LSTM model, the macro F1-score increased by 17.88%. This improvement underscores the combined strength of pre-trained transformer models like BERT and CRF layers in enhancing the performance of sequence labeling tasks.

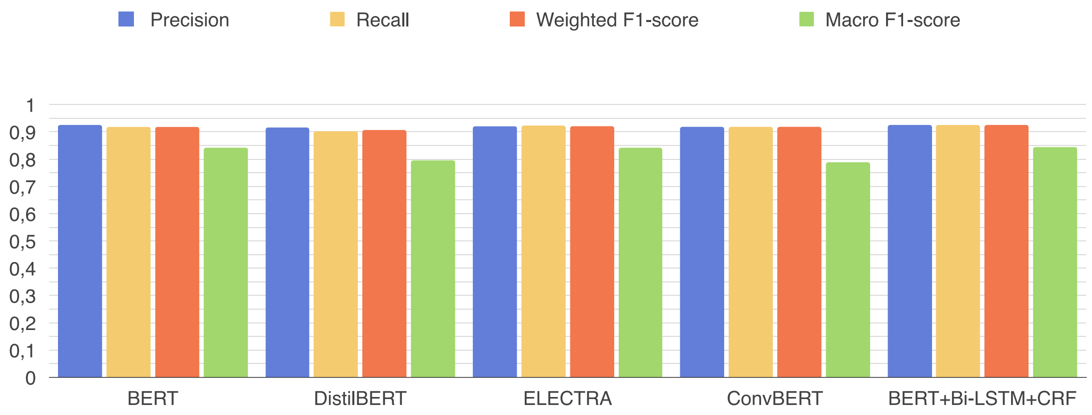

Figure 3.2. Performance of the transformers-based model according to the precision, recall, weighted and macro avg. F1-score.

Regarding resource utilization, Transformers-based models generally require higher RAM usage compared to the baseline models. Figure 3.3 shows the resource usage of transformers-based models DistilBERT stands out for its lower RAM usage compared to the other transformer models, making it a more memory-efficient option. BERT+Bi-LSTM+CRF, on the other hand, consumes significantly higher RAM due to the additional complexity introduced by the Bi-LSTM and CRF layers. The standalone ConvBERT model is also notably more resource-efficient, and still maintains a respectable level of accuracy and F1-scores, making it a viable option, especially considering its lower resource requirements.

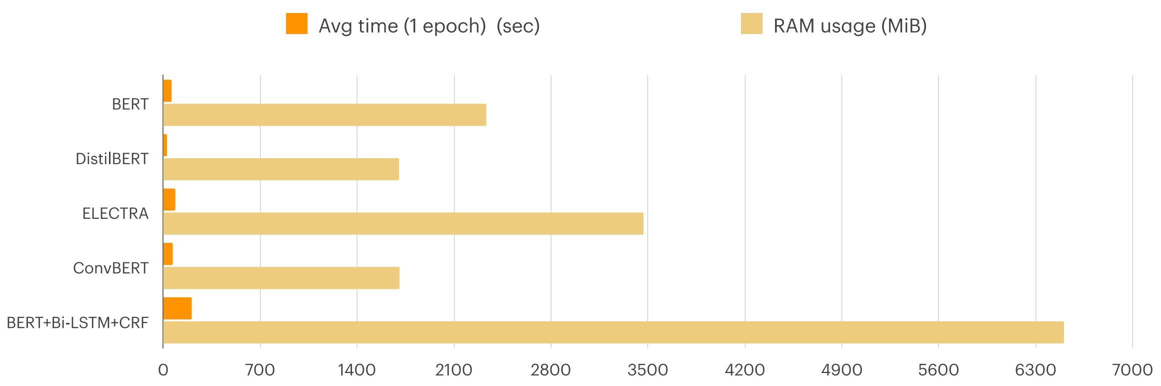

Figure 3.3. Resource usage of the transformers-based models.

DistilBERT offers the fastest training speed per epoch, making it a suitable choice for scenarios where rapid iterations are necessary. BERT+Bi-LSTM+CRF, with its higher computational demands, requires the longest time per epoch.

BERT+Bi-LSTM+CRF introduces additional complexity with the Bi-LSTM and CRF layers, potentially offering improved performance at the cost of increased model complexity and reduced interpretability. On the other hand, individual Transformer models offer more straightforward architectures, facilitating easier interpretation of model decisions while they are still powerful.

Ultimately, the preferred model depends on the specific trade-offs between performance, resource usage, training speed, and model complexity that align with the priorities of the task and available resources. For resource-constrained environments, DistilBERT and ConvBERT may offer a balanced compromise between performance and efficiency, while for tasks demanding maximal performance, BERT+Bi-LSTM+CRF and ELECTRA might be the preferred choice despite its higher resource requirements.

Table 3.5 provides a comprehensive overview of the precision, recall, and F1-score achieved by the BERT model for each entity type in the test set. We decided to analyze the results from the BERT model instead of the BERT+Bi-LSTM+CRF setup, even though the latter performed slightly better. We made this choice because the BERT model offers similar performance while using less computer memory and taking less time to train. By using the BERT model's results, we strike a balance between performance and efficiency, allowing for a thorough evaluation of our named entity recognition task given our computing constraints.

Each row in the table corresponds to a specific entity type, with the 'support' column delineating the respective count of instances for each entity within the test set. Upon reviewing the table, it's evident that the model is generally better at correctly predicting the start of named entities ('B' tags) compared to their subsequent parts ('I' tags). This tendency can be explained by the fact that there are usually more examples of 'B' tags in the training data, making it easier for the model to learn patterns associated with them. For example, entity types like 'B-category' and 'B-feature' tend to have higher prediction scores because they appear frequently in the training data. On the other hand, when there are fewer examples of an entity type in the training data, the model's performance tends to suffer. However, some entity types with low support counts, like 'B-gender,' 'B-size,' and 'B-height,' still achieve good prediction scores.

Table 3.5. Performance of the BERT model according to the accuracy, precision, recall, f1-score for each entity.

Tags	Precision	Recall	F1-score	Support
B-size	0.9718	0.9452	0.9583	73
B-height	0.9438	0.9655	0.9545	87
B-gender	1.0000	0.9268	0.9620	41
B-condition	1.0000	0.3333	0.5000	6
B-price	0.8974	0.9459	0.9211	37
B-discount	0.9545	0.4468	0.6087	47
B-category	0.9776	0.9793	0.9784	1736
B-fabric	0.8763	0.9466	0.9101	262
B-brand	0.7679	0.8776	0.8190	49
B-seasonal	0.9744	0.9268	0.9500	41
B-color	0.9721	0.9672	0.9696	396
B-website	1.0000	0.8333	0.9091	18
B-age	0.9091	0.9756	0.9412	41
B-time	0.6000	0.6667	0.6316	9
B-feature	0.9426	0.9053	0.9236	961
B-product code	0.9265	0.9764	0.9508	297
I-size	0.9571	0.9437	0.9504	71
I-height	0.6250	0.8333	0.7143	6
I-price	0.9048	0.8636	0.8837	22
I-discount	0.9231	0.6000	0.7273	20
I-category	0.7895	0.7836	0.7865	134
I-fabric	0.6667	0.1818	0.2857	11
I-brand	1.0000	1.0000	1.0000	1
I-seasonal	0.7143	1.0000	0.8333	5
I-color	0.9612	0.9429	0.9519	105
I-website	1.0000	0.8333	0.9091	6
I-age	1.0000	1.0000	1.0000	10
I-time	0.5000	1.0000	0.6667	1
I-feature	0.9448	0.9013	0.9226	304
I-product code	0.9483	0.9167	0.9322	60
O	0.5907	0.6636	0.6250	324

This is because these entity types represent clear categories (e.g., male/female for 'B-gender'), allowing the model to make accurate predictions even with limited training examples. Conversely, entity types like 'B-brand' tend to have lower prediction scores because they are less categorical in nature and have fewer examples in the training data, making it harder for the model to predict them accurately.

4. CONCLUSION

In conclusion, this study has demonstrated the effectiveness of utilizing various machine learning models for Named Entity Recognition (NER) to improve e-commerce search queries in Turkish.The Bi-LSTM model demonstrated solid performance but had inconsistent results across different classes. The Bi-LSTM+CRF model significantly improved overall performance by enhancing sequential dependency capture.

Incorporating word embedding models yielded further enhancements. The FastText+BiLSTM model excelled in dealing with out-of-vocabulary terms and varied query formats, and incorporating CRF into this framework resulted in the FastText+BiLSTM+CRF model, which outperformed other baseline models.

Transformer-based models outperformed previous models, with ELECTRA achieving the highest F1-scores. Combining pre-trained BERT with Bi-LSTM+CRF also yielded top results. DistilBERT and ConvBERT offered a good balance of performance and efficiency, while BERT+Bi-LSTM+CRF and ELECTRA were best for high-performance needs.

For future work, we aim to integrate these advanced NER models within real-world e-commerce applications. Implementing and evaluating these models in live environments will provide valuable insights into their impact on user experience, search relevance, and overall business metrics. Specifically, we will assess how accurately the models enhance product discovery, improve search query understanding, and ultimately contribute to increased customer satisfaction and business efficiency.

REFERENCES

[1] Chowdhary, G. G. (2020). Natural language processing. Fundamentals of artificial intelligence, 603-649.

[2] Chiticariu, L., Li, Y., & Reiss, F. (2013, October). Rule-based information extraction is dead! long live rule-based information extraction systems!. In Proceedings of the 2013

[3] Rabiner, Lawrence, and Biinghwang Juang. "An introduction to hidden Markov models." ieee assp magazine 3.1 (1986): 4-16.

[4] Berger, Adam, Stephen A. Della Pietra, and Vincent J. Della Pietra. "A maximum entropy approach to natural language processing." Computational linguistics 22.1 (1996): 39-71.

[5] Lafferty, John, Andrew McCallum, and Fernando CN Pereira. "Conditional random fields: Probabilistic models for segmenting and labeling sequence data." (2001).

[6] Zeroual, I., & Lakhouaja, A. (2018). Data science in light of natural language processing: An overview. Procedia Computer Science, 127, 82-91.

[7] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., & Kuksa, P. (2011). Natural language processing (almost) from scratch. Journal of machine learning research, 12(ARTICLE), 2493-2537.

[8] Huang, Z., Xu, W., & Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. arXiv preprint arXiv:1508.01991.

[9] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., & Dyer, C. (2016). Neural architectures for named entity recognition. arXiv preprint arXiv:1603.01360.

[10] Chiu, J. P., & Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. Transactions of the association for computational linguistics, 4, 357-370.

[11] Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. arXiv preprint arXiv:1705.00108.

[12] Li, J., Sun, A., Han, J., & Li, C. (2020). A survey on deep learning for named entity recognition. IEEE Transactions on Knowledge and Data Engineering, 34(1), 50-70.

[13] Mikolov, Tomas and Karafiat, Martin and Burget, Lukas and Cernocky, Jan and Khudanpur, Sanjeev. "Recurrent neural network based language model." Interspeech. Vol. 2. No. 3. 2010.

[14] Hochreiter, S. and Schmidhuber J. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[15] Chung, Junyoung and Gulcehre, Caglar and Cho, KyungHyun and Bengio, Yoshua. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).
[16] Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." Advances in neural information processing systems 27 (2014).
[17] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).
[18] Cho, Kyunghyun and Van Merrienboer, Bart and Bahdanau, Dzmitry and Bengio, Yoshua. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259 (2014).
[19] Zhou, Chunting and Sun, Chonglin and Liu, Zhiyuan and Lau, Francis. "A C-LSTM neural network for text classification." arXiv preprint arXiv:1511.08630 (2015).
conference on empirical methods in natural language processing (pp. 827-832).
[20] Liu, Gang and Jiabao Guo. "Bidirectional LSTM with attention mechanism and convolutional layer for text classification." Neurocomputing 337 (2019): 325-338.
[21] Mikolov, Tomas and Chen, Kai and Corrado, Greg and Dean, Jeffrey. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
[22] Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759.
[23] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).
[24] Wang, Bin and Wang, Angela and Chen, Fenxiao and Wang, Yuncheng and Kuo, C-C Jay. "Evaluating word embedding models: Methods and experimental results." APSIPA transactions on signal and information processing 8 (2019): e19.
[25] Rezaeinia, S. M., Rahmani, R., Ghodsi, A., & Veisi, H. (2019). Sentiment analysis based on improved pre-trained word embeddings. Expert Systems with Applications, 117, 139-147.
[26] Yu, L. C., Wang, J., Lai, K. R., & Zhang, X. (2017, September). Refining word embeddings for sentiment analysis. In Proceedings of the 2017 conference on empirical methods in natural language processing (pp. 534-539).

[27] Onan, A. (2021). Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. Concurrency and Computation: Practice and Experience, 33(23), e5909.
[28] Stein, R. A., Jaques, P. A., & Valiati, J. F. (2019). An analysis of hierarchical text classification using word embeddings. Information Sciences, 471, 216-232.
[29] Zou, W. Y., Socher, R., Cer, D., & Manning, C. D. (2013, October). Bilingual word embeddings for phrase-based machine translation. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1393-1398).
[30] Qi, Y., Sachan, D. S., Felix, M., Padmanabhan, S. J., & Neubig, G. (2018). When and why are pre-trained word embeddings useful for neural machine translation?. arXiv preprint arXiv:1804.06323.
[31] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.
[31] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.
[32] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training.
[33] Zampieri, M., Malmasi, S., Nakov, P., Rosenthal, S., Farra, N., & Kumar, R. (2019). Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). arXiv preprint arXiv:1903.08983.
[34] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders. arXiv preprint arXiv:1908.08345.
[35] Souza, F., Nogueira, R., & Lotufo, R. (2019). Portuguese named entity recognition using BERT-CRF. arXiv preprint arXiv:1909.10649.
[36] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry G., Askell A., Mishkin P., Clark J., Krueger G. & Sutskever, I. (2021, July). Learning transferable visual models from natural language supervision. In International conference on machine learning (pp. 8748-8763). PMLR.
[37] Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., ... & Sutskever, I. (2021, July). Zero-shot text-to-image generation. In International Conference on Machine Learning (pp. 8821-8831). PMLR.

[38] Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017, May). A new chatbot for customer service on social media. In Proceedings of the 2017 CHI conference on human factors in computing systems (pp. 3506-3510).
[39] Shawar, B. A., & Atwell, E. (2007). Chatbots: are they really useful?. Journal for Language Technology and Computational Linguistics, 22(1)
[40] Lende, S. P., & Raghuwanshi, M. M. (2016, February). Question answering system on education acts using NLP techniques. In 2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave) (pp. 1-6). IEEE.

[41] Abacha, A. B., & Zweigenbaum, P. (2015). MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. Information processing & management, 51(5), 570-594.

[42] Crawford, M., Khoshgoftaar, T. M., Prusa, J. D., Richter, A. N., & Al Najada, H. (2015). Survey of review spam detection using machine learning techniques. Journal of Big Data, 2(1), 1-24.
[43] Ardila, R., Branson, M., Davis, K., Henretty, M., Kohler, M., Meyer, J., ... & Weber, G. (2019). Common voice: A massively-multilingual speech corpus. arXiv preprint arXiv:1912.06670.
[44] Nassif, A. B., Shahin, I., Attili, I., Azzeh, M., & Shaalan, K. (2019). Speech recognition using deep neural networks: A systematic review. IEEE access, 7, 19143-19165.
[45] Xiong, W., Wu, L., Alleva, F., Droppo, J., Huang, X., & Stolcke, A. (2018, April). The Microsoft 2017 conversational speech recognition system. In 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 5934-5938). IEEE.
[46] Nadeau, D., & Sekine, S. (2007). A survey of named entity recognition and classification. Lingvisticae Investigationes, 30(1), 3-26.
[47] Guo, J., Xu, G., Cheng, X., & Li, H. (2009, July). Named entity recognition in query. In Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval (pp. 267-274).
[48] Petkova, D., & Croft, W. B. (2007, November). Proximity-based document representation for named entity retrieval. In Proceedings of the sixteenth ACM conference on Conference on information and knowledge management (pp. 731-740).
[49] Ritter, A., Clark, S., & Etzioni, O. (2011, July). Named entity recognition in tweets: an experimental study. In Proceedings of the 2011 conference on empirical methods in natural language processing (pp. 1524-1534).

[50] Liu, X., Zhang, S., Wei, F., & Zhou, M. (2011, June). Recognizing named entities in tweets. In Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies (pp. 359-367).
[51] Babych, B., & Hartley, A. (2003). Improving machine translation quality with automatic named entity recognition. In Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003.
[52] Luo, L., Yang, Z., Yang, P., Zhang, Y., Wang, L., Lin, H., & Wang, J. (2018). An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition. Bioinformatics, 34(8), 1381-1388.
[53] Leaman, R., & Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. In Biocomputing 2008 (pp. 652-663).
[54] Cunningham, H., Maynard, D., Bontcheva, K., and Tablan, V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications, in: Proceedings of the Meeting of the Association for Computational Linguistics, 2002.
[55] Bikel, D. M., Schwartz, R., & Weischedel, R. M. (1999). An algorithm that learns what's in a name. Machine learning, 34, 211-231.
[56] Sutton, C., & McCallum, A. (2012). An introduction to conditional random fields. Foundations and Trends® in Machine Learning, 4(4), 267-373.
[57] Finkel, J. R., Grenager, T., & Manning, C. D. (2005, June). Incorporating non-local information into information extraction systems by gibbs sampling. In Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05) (pp. 363-370).
[58] Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.
[59] Liu, P. J., Saleh, M., Pot, E., Goodrich, B., Sepassi, R., Kaiser, L., & Shazeer, N. (2018). Generating wikipedia by summarizing long sequences. arXiv preprint arXiv:1801.10198.
[60] Kitaev, N., & Klein, D. (2018). Constituency parsing with a self-attentive encoder. arXiv preprint arXiv:1805.01052.
[61] Duch, W., Setiono, R., & Zurada, J. M. (2004). Computational intelligence methods for rule-based data understanding. Proceedings of the IEEE, 92(5), 771-805.
[62] Chiticariu, L., Krishnamurthy, R., Li, Y., Reiss, F., & Vaithyanathan, S. (2010, October). Domain adaptation of rule-based annotators for named-entity recognition tasks.

In Proceedings of the 2010 conference on empirical methods in natural language processing (pp. 1002-1012).
[63] McCallum, A., & Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons.
[64] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436-444.
[65] Waldrop, M. M. (2019). What are the limits of deep learning?. Proceedings of the National Academy of Sciences, 116(4), 1074-1077.
[66] Pakhale, K. (2023). Comprehensive overview of named Entity Recognition: Models, Domain-Specific applications and challenges. arXiv preprint arXiv:2309.14084.
[67] Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2022). Pre-trained language models and their applications. Engineering.
[68] Liu, X., & Wang, C. (2021). An Empirical Study on Hyperparameter Optimization for Fine-Tuning Pre-trained Language Models. arXiv preprint arXiv:2106.09204.
[69] Güneş, A., & Tantuğ, A. C. (2018, May). Turkish named entity recognition with deep learning. In 2018 26th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). IEEE.
[70] Tür, G., Hakkani-Tür, D., & Oflazer, K. (2003). A statistical information extraction system for Turkish. Natural Language Engineering, 9(2), 181-210.
[71] Seker, G. A., & Eryigit, G. (2017). Extending a CRF-based named entity recognition model for Turkish well formed text and user generated content. Semantic Web, 8(5), 625-642.
[72] Akkaya, E. K., & Can, B. (2021). Transfer learning for Turkish named entity recognition on noisy text. Natural Language Engineering, 27(1), 35-64.
[73] Akdemir, A. (2018). Named Entity Recognition in Turkish Using Deep Learning Methods and Joint Learning (Doctoral dissertation, PhD Thesis, Bogaziçi University).
[74] Çelikmasat, G., Aktürk, M. E., Ertunç, Y. E., Issifu, A. M., & Ganiz, M. C. (2022, August). Biomedical Named Entity Recognition Using Transformers with biLSTM+ CRF and Graph Convolutional Neural Networks. In 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA) (pp. 1-6). IEEE.
[75] Dinç, D. (2022). Financial named entity recognition for Turkish news texts (Master's thesis, Middle East Technical University).
[76] Çarık, B. (2022). Enhancing named entity recognition in Turkish by ıntegrating external knowledge and extra layers into transformer-based models (Doctoral dissertation).

[77] Topçu, B., & El-Kahlout, I. D. (2021, September). TR-SEQ: Named Entity Recognition Dataset for Turkish Search Engine Queries. In Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021) (pp. 1417-1422).
[78] Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. arXiv preprint arXiv:2003.10555.
[79] Bhange, B. R., Chengy, X., Bowden, M., Goyaly, P., Packery, T., & Javedy, F. (2020). Named Entity Recognition for E-Commerce Search Queries.
[80] Zhang, H., Hennig, L., Alt, C., Hu, C., Meng, Y., & Wang, C. (2020). Bootstrapping named entity recognition in e-commerce with positive unlabeled learning. arXiv preprint arXiv:2005.11075.
[81] Wen, M., Vasthimal, D. K., Lu, A., Wang, T., & Guo, A. (2019, December). Building large-scale deep learning system for entity recognition in e-commerce search. In Proceedings of the 6th IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 149-154).
[82] Siddiqa, A., Banerjee, S., & Garera, N. (2022). Semi-supervised Named Entity Recognition to solve label scarcity challenges for E-Commerce use-cases. KDD EcomGen.
[83] Graves, A., Santiago F., and Jürgen S., "Bidirectional LSTM networks for improved phoneme classification and recognition." International conference on artificial neural networks. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
[84] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. Transactions of the association for computational linguistics, 5, 135-146.
[85] Ozdemir, A., & Yeniterzi, R. (2020, December). Su-nlp at semeval-2020 task 12: Offensive language identification in turkish tweets. In Proceedings of the Fourteenth Workshop on Semantic Evaluation (pp. 2171-2176).
[86] Stefan, S. (2020). Berturk - bert models for turkish.
[87] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
[88] Jiang, Z. H., Yu, W., Zhou, D., Chen, Y., Feng, J., & Yan, S. (2020). Convbert: Improving bert with span-based dynamic convolution. Advances in Neural Information Processing Systems, 33, 12837-12848.

[89] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., ... & Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144.

[90] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[91] Kudo, T. (2018). Subword regularization: Improving neural network translation models with multiple subword candidates. arXiv preprint arXiv:1804.10959.

[92] Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226.

[93] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[94] Köksal, A., (2018), Github, https://github.com/akoksal/Turkish-Word2Vec, 15.05.2024. Internet

[95] Güngör, Onur, and Eray Yıldız. "Word Embeddings." (2017).

[96] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation*.

[97] Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

ÖZGEÇMİŞ

Beyzanur Saraçlar

E-ticaret sektörüne yönelik veri analitiği ve yapay zeka uygulamalarında tecrübe sahibi bir veri bilimciyim. Özellikle doğal dil işleme alanında çalışarak, makine öğrenimi ve derin öğrenme modelleri üzerinde çalışmalar yaptım. Bu alanlarda e-ticaret sektöründeki sorunlara çözüm sunmaya yönelik projelerde yer aldım. Araştırma ilgi alanlarım arasında arama motoru reklamcılığı, kullanıcı yanıtı tahmini ve makine öğrenimi ile reklam kreatiflerinin otomatik oluşturulması gibi konular bulunmaktadır.

Deneyim

2019 – Şu an

Sefamerve (Ar-Ge Merkezi) | Maltepe, İstanbul

- Veri toplama, hazırlama, modelleme, değerlendirme ve dağıtım süreçlerini kapsayan uçtan uca analiz süreçlerini gerçekleştirdim.
- E-ticaret sektöründeki sorunları çözmek için gelişmiş makine öğrenimi (ML), derin öğrenme (DL) ve doğal dil işleme (NLP) modellerini kullanarak yapay zeka hizmetleri geliştirdim, devreye aldım ve bakımını yaptım.
- Doğal dil işleme alanında çeşitli makine öğrenimi ve yapay zeka projelerine katıldım.
- Ürün fiyat tahmini, arama sorguları için adlandırılmış varlık tanıma servisi, sorgu-sayfa alaka düzeyi skoru hesaplama ve insan kaynakları chatbot hizmeti gibi projelere katkıda bulundum.
- Bilgisayarla görme ve görüntü işleme tabanlı yapay zeka projeleri üzerinde çalışan takım arkadaşlarına destek vererek bu alanda değerli deneyimler kazandım.
- Çalışmalarımızı farklı uluslararası konferanslarda sunduk ve ulusal ve uluslararası dergilerde yayımladık.

2015 – 2017

İstanbul Şehir Üniversitesi (Veri Bilimi Laboratuvarı)

- Arama motoru reklamcılığı alanında kapsamlı çalışmalar yaptım.
- Kullanıcı yanıtlarını tahmin ettim, özellikle Tıklama Oranı (CTR) ve Dönüşüm Oranı (CR) üzerine yoğunlaştım.
- Makine öğrenimi ve derin öğrenme tekniklerini kullanarak reklam kreatiflerinin otomatik olarak oluşturulmasını inceledim.