EXTRACTING TOKEN-LEVEL SEMANTIC MATCHING IN TEXT-PAIR CLASSIFICATION TASKS

A Dissertation Presented

by

YOUNGWOO KIM

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

April 2024

Robert and Donna Manning College of
Information and Computer Sciences

EXTRACTING TOKEN-LEVEL SEMANTIC MATCHING IN TEXT-PAIR CLASSIFICATION TASKS

A Dissertation Presented

by

YOUNGWOO KIM

Approved as to style and content by:

James Allan, Chair

Mohit Iyyer, Member

Razieh Rahimi, Member

Rajesh Bhatt, Member

Ramesh Sitaraman, Associate Dean for
Educational Programs and Teaching
Robert and Donna Manning College of
Information and Computer Sciences

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to all those who have supported me throughout my PhD journey and contributed to the successful completion of this dissertation. This work would not have been possible without the guidance, encouragement, and assistance of numerous individuals, entities, and organizations.

Academic Mentors and Advisors

First of all, I'd like to thank James Allan for his invaluable advice throughout the dissertation writing and contributions as a committee chair. Under James's supervision, I have learned the philosophies of research, leadership, and the art of navigating the academic landscape. Moreover, I am truly grateful for the freedom James has granted me in exploring various research directions and for his generous financial support.

I would also like to acknowledge the advice I received from Professor Negin Rahimi. As a coauthor of four conference papers, she provided an indispensable role in the research project. Three out of four chapters in this dissertation are highly influenced by these papers; her guidance and contributions to the help are enormous.

I also would like to thank Mohit Iyyer and Rajesh Bhatt for being committee members. Mohit's suggestions and constructive critique have been critical in deciding the major directions of the dissertation.

I would like to express my sincere gratitude to Myungha Jang for being an outstanding mentor during my first two years as a Ph.D. student. Her guidance and support in writing my first paper were invaluable, and her contributions played a significant role in shaping the work. Although not listed as a co-author, her dedication and expertise throughout the research process should be acknowledged and appreciated.

I would like to express my appreciation to the members of the NLP reading group for fostering a stimulating environment where we shared research philosophies, discussed the latest academic papers, and exchanged valuable writing advice. Their high expectations and enthusiasm as researchers were truly inspiring and motivated me to strive for excellence in my own work.

I am grateful for the guidance and support provided by my internship mentors. During my internship at CodaMetrix, I acquired a robust knowledge of ML fundamentals for industry from the mentor Cheng Li.

I would like to express my gratitude to those who supported me during my internship at Facebook (now Meta). My mentor, Fei Jia, provided me with warm-hearted and sincere advice throughout my time there. Under Fei Jia's guidance, I gained valuable insights into working philosophies, effective utilization of milestones, and professional communication strategies. These lessons have had a lasting impact on my approach to work and collaboration.

I would like to extend a special thanks to Nicole Espinosa, my recruiter, for her exceptional support during a challenging time. When I expressed concerns about my assigned role and considered leaving the internship, Nicole went above and beyond to help me find a new team that better aligned with my interests and goals. Her dedication and understanding were instrumental in transforming my internship experience.

Looking back, I can confidently say that the three months I spent as an intern at Facebook were among the most exciting and enriching periods of my Ph.D. journey. The knowledge, skills, and relationships I developed during this time have had a profound influence on my personal and professional growth.

I extend my gratitude to the anonymous reviewers who dedicated their time and expertise to provide constructive feedback on my papers. Their insightful comments and suggestions not only helped improve the quality of my work but also taught me valuable

lessons on how to effectively structure and present research findings. I am truly grateful for their contributions to my growth as a researcher.

I would also like to acknowledge the contributions from AI systems and their providers, ChatGPT from OpenAI and Claude 3 from Anthropic, for their assistance during my research and writings. Their responsiveness combined with high quality responses were beyond what any human can handle. Whenever I was stuck with writing or research I asked for advice from AIs and they gave me broadly generic, but thorough descriptions.

Specifically, AI assistance was used for these purposes: revising texts, identifying grammar errors and typos, composing a draft from bullet points, drawing graphs, inquiries about LaTeX functions, asking for what is missing, generic questions about roles of each section, enriching summary sections and emotional supports.

Lab mates and staffs

I enjoyed the work environment at CIIR and I thank my lab mates and colleagues for making my time at CIIR wonderful. I would like to express my gratitude to my senior lab mates, John Foley, Jiepu Jiang, Liu Yang, Myungha Jang, Qingyao Ai, Daniel Cohen, and Hamed Zamani, for being role models and sharing their knowledge through their teaching.

To my fellow generations, Shiekh Sarwar, Hamed Bonab, Shahrzad Naseri, Ali Montazeralghaem, Helia Hashemi, Chen Qu, Yen-Chieh Lien, Lakshmi Vikraman, Puxuan Yu, Zhiqi Huang, Tanya Chowdhury, and Nazanin Jafari, who started at similar times and shared the academic growth and experienced challenges like the pandemic together, I am thankful for your camaraderie and support.

I'd liked thank new generation PhD students, Christos Samarinas, Yaxin Zhu, Hansi Zeng, Alireza Salemi, Mahta Rafiee, and Julian Killingback, for bringing new energetic spirits to CIIR. They introduced me to new technologies, research problems and methodologies.

I would like to extend my sincere thanks to the CIIR and CSCF staff, including Jean Joyce, Kate Morruzzi, Dan Parker, Glenn Stowell, Gregory Brooks, Leeanne Leclerc, and Eileen Hamel for their invaluable administrative support throughout my time at the university. I would also like to express my special gratitude to Jean Joyce for her assistance with the numerous IRB review requests. Her guidance and support were instrumental in navigating the complex review process smoothly. Furthermore, I want to acknowledge the efforts of Dan Parker and other CSCF staff members who diligently handled various issues arising from security risks originating from my desktop. I sincerely apologize for any inconvenience caused and deeply appreciate their prompt and professional response in resolving these matters.

Friends, Family and Personal support

Finally, but most importantly, I would like to thank my friends and family for bearing with me on this difficult journey.

I was fortunate to know those Korean CS grad friends, Myungha Jang, Shinyoung Cho, Souyoung Jin, Ronald Seo, Eunjeong Hwang, Sunjae Kwon, and Hochul Hwang, with whom I could share concerns about research and work, and from whom I received emotional and practical support.

I was also fortunate to have great friends who not only supported me during the challenging times of the pandemic and the demanding periods of my PhD journey but also brought joy and laughter into my life. Jiah Lee, Tyler Seabury, Mina Lee, Deukjae Lee, Minji Lee, Juhyeon Lee, Haknyeong Hong, Taehyun Kim, O-sung Kwon, Heejoong Choi, Jungwon Kyung, and Zhipeng Tang were the friends who stood by me, offering their unwavering support and creating memorable experiences that I will always treasure.

I attribute all my successes, including this Ph.D., to my parents. My parents have allowed me to grow independently, supporting me when I was in trouble, but allowing me

to make my own decisions. I also want to thank my elder brother, who provided support for my journey.

I would like to express my heartfelt gratitude to the Kwanjeong Educational Foundation for placing their trust in me and supporting my academic journey through their scholarship. This recognition not only provided financial assistance but also instilled in me a greater sense of self-confidence and belief in my own abilities.

Grants

This work was supported in part by the Center for Intelligent Information Retrieval, in part by NSF grant #2106282, in part by NSF grant #1813662, and in part by NSF grant #1819477. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

ABSTRACT

EXTRACTING TOKEN-LEVEL SEMANTIC MATCHING IN TEXT-PAIR CLASSIFICATION TASKS

APRIL 2024

YOUNGWOO KIM

B.Sc., POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

M.Sc., UNIVERSITY OF MASSACHUSETTS AMHERST

Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor James Allan

This dissertation presents approaches to obtain interpretability and extract token-level semantics from transformer-based text-pair classification models. We focus on both token-level task-solving and model explanation for natural language inference (NLI) and information retrieval tasks. We hypothesize that if these models can successfully address text-pair classification problems, they must inherently possess the capacity to solve corresponding token-level problems to a certain degree. However, the objective is not only to transform the text-pair classification solutions into token-level inferences that are prerequisite for these tasks but also to explain the decision-making processes and behavior of these models.

The first half of the dissertation comprises two parts focused on deriving interpretability from neural NLI models. The initial part proposes a sequence labeling task called classification role labeling (CRL) to represent token-level semantic understanding in NLI. The

goal is to label each token in the text-pair based on their semantic alignment and whether they contribute to contradictions. We show that such sequence labeling models can be trained by weak-supervision from a NLI classification model. The subsequent part studies the use of CRL for explaining contradictory claims from biomedical articles, demonstrating the effectiveness of our novel model, PAT, on the Cond-NLI dataset.

The second half of the dissertation spans two parts targeting the ad-hoc retrieval task, specifically on explaining the mechanism behind query-document relevance scoring functions. One part investigates local alignment rationales for explaining query-document relevance classification from a black-box model, proposing perturbation-based metrics to evaluate alignment rationale quality. In the other part, we provide global explanations for neural ranking models, by representing their semantic matching behavior as "relevance thesaurus" containing semantically related query-term and document-term pairs. This thesaurus can reveal corpus-specific features and biases, supporting the utility of our explanation method. Overall, this four-part dissertation introduces novel approaches to complement interpretability in neural text-pair classification models, extracting token-level semantics and alignment rationales without the need for additional human annotations, while also providing insights into the models' decision-making processes.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Background

1.1.1 Text-pair classification

There are natural language processing (NLP) tasks that are categorized as text-pair classification tasks. For example, document retrieval tasks can be considered as a **query-document relevance classification**, where one input text is a query string and another input text is a document and the probability of the relevant class is used to rank documents (Nallapati, 2004).

Natural language inference (NLI) is another example (Williams et al., 2018). In NLI, one input text is called a premise and the other is called a hypothesis. The goal is to classify the pair into one of three classes based on whether the premise entails the hypothesis, it contradicts the hypothesis or neither case is implied. The definitions and some examples are in Table 1.1.

While each text-pair classification task has different definitions, they are similar in a way that the decision should be made by checking how the meaning indicated by one text appears in the other text. In this dissertation, we mainly focus on NLI and query-document relevance classification, hoping that the conclusions can later be generalized into other text-pair classification tasks, such as fact checking (Thorne et al., 2018) and semantic textual similarity (Agirre et al., 2013).

Solutions for text-pair classification tasks have procedures that compare tokens in one text to the tokens in the other text. Earlier solutions for text-pair classification contain procedures that check if each of the tokens in one text can be matched to some tokens in the

Premise	Label	Hypothesis
The great thing is to keep calm." Julius groaned.	entailment	Julius made a groaning sound.
yeah i mean just when uh the they military paid for her education	contradiction	The military didn't pay for her education.
uh-huh well I've enjoyed talking to you.	neutral	I liked talking to you about sports.

Table 1.1: Three example sentence pairs from MultiNLI dataset with the corresponding classification labels. **Entailment** class indicates that the hypothesis is entailed (can be inferred) from the premise. **Contradiction** class indicates that the premise and hypothesis are contradictory and cannot be true at the same time. **Neutral** class indicates that the hypothesis neither can be inferred nor contradicts the premise.

other text and build decisions based on such matches. In query-document relevance classification, the query text specifies what is required to be relevant. Approaches like BM25 or query-likelihood score each document based on the query term frequency in the document and among other statistics. In NLI, one strategy is to build a sequence of edits that transforms the premise text into the hypothesis text and compute scores based on the edit sequence (MacCartney and Manning, 2007). Other strategies for both tasks also include computing overlap between two texts, often combined with n-gram features, synonym or ontology dictionary and vector similarity of the tokens (Malakasiotis and Androutsopoulos, 2007). For example, in the neutral labeled example shown in Table 1.1, the phrase "I liked talking to you" in the hypothesis can be aligned to " I've enjoyed talking to you", in the premise, as each token in them can be paired with one another. However, the phrase "about sports" in the hypothesis cannot be aligned to any tokens in the premise that entail its meaning. Thus, the text pair can be classified as neutral. Overall, there exist tractable connections between the tokens of one text to the tokens in the other that either positively or negatively contribute to the classification decision.

In the recent deep neural network based approaches for text-pair classifications, the matching between two texts has become implicit and intractable to pin down. Transformer architecture (Vaswani et al., 2017) is the dominant and often the most effective method to solve text-pair classification tasks. Attention mechanism is the key of the Transformer

architecture, which allows vector representation for each token to be computed with other vectors with dynamically learned direction and weights. Transformer is composed of multiple Transformer blocks, and at each Transformer block, there is one vector that corresponds to each token.

These vectors are processed to incorporate contextual information from the entire sequence. The mechanism responsible for this contextual integration is the self-attention mechanism. It computes attention scores that determine the relevance of all other tokens in the sequence for a given token, allowing each token's representation to be dynamically updated with information from the whole sequence. Each token's vector representation is combined with vectors derived from other tokens in the sequence. This combined representation is then passed through feed-forward layers and residual connections, which output the vector representations for the corresponding tokens in the next transformer block.

A cross-encoder is one popular way of solving text-pair classification with Transformer, in which the tokens from two texts are concatenated and fed into Transformer and token embeddings for these tokens are summed with segment embeddings so that tokens from different texts can be differentiated.

In cross-encoder, the vectors from each token are "mixed" with many other tokens' vectors in multiple layers. Thus, the attribution, or the matching cannot be simply represented. In another strategy called bi-encoder, each of the two texts is encoded into a single vector separately from the other. Then the classification decision is built based on two vectors from the two texts (Yu et al., 2021). Again, each token's vectors are combined by Transformer and token-level matching is not explicitly captured.

1.1.2 Motivations

In these neural approaches, term matching and comparisons of token semantics are instantiated as computations on vectors, which generally yield accurate results. However, the underlying mechanisms of these models cannot provide explanations for how or why a

Question: Does treatment with antihypertensives protect against cardiovascular incidents? **Claim1:** A antihypertensives drug doxazosin prevents cardiovascular morbidity . [Answer: **Yes**] **Claim2:** A antihypertensives drug losartan increases the risk of congestive heart failure. [Answer: **No**]

Table 1.2: A simplified example of contradictory claims from BioClaim dataset. The question represents the potential information need. Two claims are showing opposite outcomes (red) toward the questions, while having different conditions (yellow).

match is determined. Although Transformer-based strategies do not require explicit modeling of token-level matching to solve text-pair classification tasks, the matching information and match-attributed remain valuable for applications that build upon these tasks.

For example, search engines assist users in judging relevance on search engine result pages by displaying matched terms in the document snippets and indicating missing query terms if any, even if they do not explicitly model the token-level matching (Sarwar et al., 2021). This highlighting of matched terms and identification of missing query terms provide users with valuable insights into the relevance of the search results, facilitating their decision-making process when selecting the most appropriate document to fulfill their information needs.

Consider a case where we apply NLI to identify contradictory findings among medicine related claims. In Table 1.2, the question represents the potential information needed and two (simplified) claims are showing opposite results toward the question. While strictly speaking, these two claims are not contradictory, as they discuss different medicines and symptoms, classifying this pair as neutral and considering them completely unrelated may not be ideal, given the context of the information needed. A practical application should identify the contrasting aspects and differing conditions to provide a comprehensive range of answers, acknowledging the nuanced relationship between the two claims.

Such fine-grained matching information would also be useful to apply adhoc fixes to the neural models. For example, a dictionary of synonyms may be available, but augmenting

them into existing neural models could be challenging or costly. Consider a scenario where the query is "Where is CIKM 2022" and the candidate documents contain the location for "CIKM 2022", but only mention "The Conference on Information and Knowledge Management", without including the abbreviation "CIKM". In this case, post-processing can be applied to overwrite the neural model's decision when it fails to recognize the synonyms.

In this dissertation, our goal is to address token-level NLP tasks, such as extracting matching information, which are closely or causally related to text-pair classification tasks. *It's critical to highlight that we exclude the use of additional token-level annotations.* Instead, we rely on the models or training data from text-pair classification tasks to tackle the token-level tasks.

There are three reasons behind the choice to explain the targeted text-pair classification model using token-level matching extracted from the model itself: faithfulness to the model, cost-effectiveness of annotations, and ambiguity in annotation criteria.

First, if our goal is to explain the targeted text-pair classification model, it is unlikely that manual annotations that are built *independent* of the target model are the best signal to explain the targeted model. Moreover, the process of developing a system that extracts token-level matching from text-pair tasks can provide valuable insights about the behaviors or weaknesses of the models.

Second, text-pair level annotations are easier to collect than the token-level annotations. In many applications, text-pair level annotations could be extracted from users' behaviors. For example, in search engines, relevance between query and documents can be inferred from users' click records (Jung et al., 2007). However, few applications give token-level matching information. Moreover, text-pair annotations are already available in many applications and domains while token-level ones are not.

Additionally, it is common for texts to have ambiguous criteria regarding which tokens should be included in the match. There are cases where even when a query term and a document term contain overlapping concepts, additional context is necessary to assign a

relevance score to the pair. There is no clear criterion whether and when to include context tokens. Moreover, the degree of relation between terms can vary widely, demanding a distinction between terms that are nearly synonymous and those that are only topically related.

1.2 Contributions

We hypothesize that if neural text-pair classification models can successfully address text-pair classification problems, they must inherently possess the capacity to solve corresponding token-level problems to a certain degree. The objective of this dissertation is to discover methodology to transform the transformer-based text-pair classification solutions into token-level inferences that are prerequisite for the text-pair tasks.

In the first half of the dissertation (chapter 3 and chapter 4), we aim at deriving interpretability from neural natural language inference (NLI) models. Specifically, beyond the overall NLI classifications, we target identifying how each token's role differs in contributing to NLI decisions. In these chapters, we evaluate our methods based on human-labeled data, seeking plausible explanations.

In the second half of the dissertation (chapter 5 and chapter 6), we target the adhoc retrieval task, specifically on explaining the mechanism behind query-document relevance scoring functions. The goal in these chapters is to explain the given text-pair classifica-

Table 1.3: Three example sentence pairs from MNLI dataset with the corresponding classification labels (entailment, contradiction and neutral) and token-level tags : conflict (red), match (blue) and mismatch (yellow). In each row, the text on the left corresponds to the premise and the text on the right corresponds to the hypothesis.

Premise	Label	Hypothesis
The great thing is to keep calm." Julius groaned.	entailment	Julius made a groaning sound.
yeah i mean just when uh the they military paid for her education	contradiction	The military didn't pay for her education.
uh-huh well I've enjoyed talking to you.	neutral	I liked talking to you about sports.

tion model by discovering and utilizing token-level or term-level matching. The proposed methods are evaluated based on their faithfulness to the targeted models.

1.2.1 Classification role labeling and model explanations for NLI

We consider token-level inference as an intermediate step for NLI. Most existing explanations for text-classification tasks focus on identifying tokens that are considered "important" for the given prediction, without providing a detailed understanding of their specific roles in the classification process. In the contradiction example in Table 1.1, "paid", "pay", and "didn't" are all important tokens but they are important for different reasons. "paid" and "pay" are important because they refer to the same concept. "didn't" is important because it indicates the opposite outcome. These examples demonstrate the need for a more nuanced approach to token-level explanations that can capture the different roles played by each token in the classification process.

In chapter 3, the explanations for token-level matching, we define three labels to express three different types of importance (examples in Table 1.3). One way to model entailment is to consider each token in the hypothesis as a unit of information and check if that information is entailed by the premise. Following this procedure, we can define a token-level task that determines whether each token in the hypothesis is entailed by the premise. To model contradiction, we can define a task that determines whether each token represents a contradictory aspect to the other text . We name these token-level tasks classification role labeling (CRL), as they can be used to explain and differentiate how each token contributes to the classification.

Although CRL requires a more fine-grained explanation of token importance, existing neural network explanation methods can still be applied to CRL. These methods can identify important tokens for each of the classification decisions (entailment, neutral, and contradiction), which are correlated with the three CRL tags defined earlier.

These explanation methods typically calculate importance scores of input features using one or a combination of three types of signals: (1) change in outputs with respect to input perturbation (Ribeiro et al., 2016b), (2) gradients of the outputs with respect to inputs (Zeiler and Fergus, 2014; Zintgraf et al., 2017; Sundararajan et al., 2017), and (3) activated weights in the neural network (Arras et al., 2017b). While these methods do not incur additional annotation effort, we found that many of them are too computationally inefficient to be used in production deployment settings. Moreover, most methods are only investigated in very generic settings that are not specific to a particular task or architecture, and thus there should be much room to improve the accuracy by specializing on the particular problem and models.

In the model explanation area, another option is an explanation generator, which is another model that generates token-level (Lei et al., 2016) or text-format explanations (Camburu et al., 2018). Explanation generators are computationally efficient as they do not require multiple perturbations. In this study, we aim at solving CRL by building an explanation generator which explains the text-pair classification model's behaviors.

The training objective of the explanation generator is to predict the model's output changes in response to perturbations. Specifically, given a sentence pair, a number of tokens are randomly removed and the changes in the classification outcomes are measured. Tokens that cause larger changes in the classification when removed are taken as a weak supervision signal, and the sequence labeling model is trained to generate scores that can predict the model's behavior in response to perturbations.

The contributions of this work are as follows:

C1 We propose a novel weakly supervised method to train a CRL model without using any additional human-labeled data. (section 3.3)

C2 we propose utilizing the NLI model's hidden variables for token-level explanations by building a multi-task learning model that simultaneously predicts both the original

Question: Does treatment with antihypertensives protect against cardiovascular incidents? **Claim1:** A antihypertensives drug doxazosin prevents cardiovascular morbidity . [Answer: **Yes**] **Claim2:** A antihypertensives drug losartan increases the risk of congestive heart failure. [Answer: **No**]

Table 1.4: A simplified example of contradictory claims from BioClaim dataset. The question represents the potential information need. Two claims are showing opposite outcomes (red) toward the questions, while having different conditions (yellow).

NLI classification and the token-level CRL tags, which improves the model performance. (section 3.4)

C3 We show that our method, when tested on token-level annotations in the MultiNLI (Williams et al., 2018) and SNLI datasets (Camburu et al., 2018), is not only more computationally efficient than perturbation methods but also demonstrates greater precision compared to a number of strong baselines. (section 3.4)

1.2.2 Conditional Natural Language Inference using Classification Role Labeling

In chapter 4, we further investigate the sequence tagging task, classification role labeling (CRL) which was introduced in chapter 3. We focus on the scenario where we want to explain apparently contradictory claims, such as the example shown in Table 1.4, where one claim suggests a benefit of a particular treatment for a symptom, while the other indicates the opposite outcome. We formalize this as the task of identifying tokens that indicates opposite outcomes and different conditions, which we name conditional natural language inference (Cond-NLI). Cond-NLI is similar to CRL on natural language inference (NLI), as both require identifying tokens that could cause the sentence pairs to be classified as entailment, neutral, or contradiction.

We developed the BioClaim dataset specifically for the Cond-NLI task. In our investigation, we discovered that the perturbation based explanation methods are less effective with the BioClaim dataset. The reason for this is twofold. Firstly, in a text pair where only a

few tokens are responsible for a neutral classification (as in the case of MNLITag), deleting these tokens mostly results in a decision reversal. However, in scenarios where numerous tokens are not entailed by the premise (as in the case of BioClaim), perturbing individual neutral tokens in the hypothesis has minimal impact on the overall output. Secondly, when text pairs contain contradictory information, the presence of non-entailed tokens, which might otherwise lead to a neutral classification, becomes less influential, as the existence of both contradiction and not entailed information is considered contradiction.

This leads us to propose a new model, PAT (Partial ATtention), which can effectively address the Cond-NLI task. The key intuition behind PAT's effectiveness lies in its ability to generate intermediate labels for partial observations of the text pair. Unlike typical NLI models, PAT's intermediate labels can reveal partially entailed tokens and the existence of neutral tokens even when contradiction is present. By attributing decisions made at the text-pair level to individual tokens through these intermediate labels, PAT provides a more fine-grained understanding of the entailment relationships within the text pairs. This enables PAT to better handle scenarios with numerous non-entailed tokens and cases where contradictory information coexists with non-entailed tokens.

The contributions of this work are as follows:

C4 We built the BioClaim dataset, featuring challenging real-world contradictory text pairs with the token-level entailment and contradiction classifications task (Cond-NLI). (section 4.1)

C5 We demonstrate the limitations of existing model explanation methods in addressing the Cond-NLI task. (section 4.3)

C6 We introduce the novel PAT model, designed to be trained using text-pair level labels while capable of building token-level predictions. (section 4.2)

1.2.3 Alignment rationale in query-document relevance classification

In chapter 5, we investigate the task of building alignment rationales that best explain the given query-document relevance classifier, especially on how we can evaluate the given alignment rationale.

There have been efforts to explain black-box models' behavior in terms of the input features (e.g., tokens in document ranking), either by assigning importance scores to the features or selecting a subset of features that are important to preserve the models decisions (Singh and Anand, 2018; Hase et al., 2021; Fernando et al., 2019; Kim et al., 2020).

Query:	Where is SIGIR 2022
Document:	SIGIR 2022 will be held in Madrid

Table 1.5: Example alignments for the query spans 'Where is' and 'SIGIR 2022'.

However, we found few works that answer the alignment question: "If certain document tokens are important for relevance to the query, which part of the query do they correspond to?" Table 1.5 illustrates the goal of alignment. When exact match or soft match based ranking models were used, the alignment between query tokens and document tokens could be acquired with little additional effort. Such alignment information also has been used to provide more information to users, such as summarizing and visualizing each of query terms' appearances in long document (Hearst, 1995; Hoeber and Yang, 2006), also demonstrating the important of this alignment issue.

Acquiring alignment has two approaches: (1) aiming at building (ideally) 'correct' or useful alignments regardless of query-document scoring model, or (2) seeking an alignment that best explains (is faithful to) the model. We target the second approach here.

We investigate the possible uses of end-to-end input perturbation approaches, which make no assumption about the model's internal architecture. If the model outputs different decisions for a perturbed instance (a small change to the inputs), we can expect that the changed features are somehow responsible for the model decisions (Carton et al., 2020).

To expand feature importance to alignment explanation, one can test if the importance of some document tokens depends on the existence of certain query terms. This can be achieved by comparing perturbation outcomes with and without a particular query term. Unfortunately such complex perturbation is more likely to bring undesired consequences such as making the perturbed input text ungrammatical (Hase et al., 2021) or changing its meaning drastically such that the model's decision changes more than we would expect given small perturbations.

The contributions of this work are as follows:

C7 We propose perturbation-based metrics for evaluating alignment rationale in query-document relevance (section 5.2).

C8 We investigate the behavior of the proposed metrics and demonstrate that they are mostly not strong enough to make binary decisions on alignment quality (good or bad), but they can be used to rank two alignment models (section 5.3).

C9 We propose that building perturbed instances that are more comparable to the instance being explain, is the key to improvement of evaluation metrics (section 5.2). We showed that a simple approach to get more comparable instances increases the metric coverage from 13% to 68% [1] (section 5.3).

1.2.4 Global Explanation of Retrieval Models by Relevance Thesaurus

Chapter 6 shifts the focus from local to global explanation. In the previous chapters, the matching is built on the spans of the given texts, thus dependent on the tokens in the remaining context. In contrast, chapter 6's matching is at the vocabulary level, where the goal is to find the relevant query term and document term pairs that are strong indicator of relevance independent of the contexts.

[1] Based on binary-necessity category.

Fine-tuning BERT on MS MARCO (Nguyen et al., 2016a) is a very popular approach to train retrieval models (Dai and Callan, 2019; MacAvaney et al., 2019). Given the high accuracy of these models, we can expect that they capture semantic matches, such as a query term "car" being matched to a document term "vehicle". However, there is currently no systematic method for representing and analyzing the semantic matching patterns of these models, which can allow people to expect which terms are associated with a particular token. Consequently, researchers face challenges in predicting when a model might fail, such as in cases where the model incorrectly matches a query term to irrelevant document terms.

Another potential risk associated with ranking models is unintended bias in model's behavior toward certain entities or groups. For example, while it is appropriate for a model to associate the query term "car" with various car brand names (e.g., Ford, Toyota, or Honda), it may not be desirable for the model to exhibit a strong preference for a particular brand. Such bias could lead to the model favoring one brand over another when all other factors are equal, potentially resulting in unfair or skewed search results.

Query Term	Document Term	Score
injury	injure	0.26
injury	wound	0.24
car	vehiclesâ	0.68
car	ford	0.38
car	honda	0.28
cud	cuda	0.50
course	courseâ	0.78
course	coursework	0.53
when	24th	0.33
when	2002	0.22
when	2014	0.02

Table 1.6: Example entries from our relevance thesaurus.

This chapter aims at constructing global explanations for ranking models that summarize the model's semantic matching patterns to understand the models and help identify

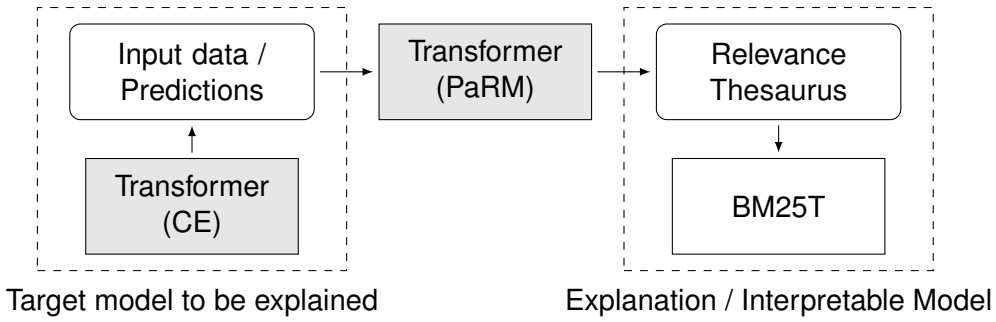


Figure 1.1: Process of how relevance thesaurus is built from the proposed method. The gray rectangles represents black-box models, while the white rectangle (BM25T) is an interpretable model.

potential risks. Our global explanation is composed of relevant pairs of query terms and document terms. We refer to this format of explanation as a **relevance thesaurus**, as illustrated in Table 1.6. That table indicates that if a query contains the term "injury" then it is likely to correspond to document terms "injure," or "wound," with the first more likely, allowing a retrieval's results to be explained directly using the term pairs.

We propose a novel method to construct a relevance thesaurus (Figure 1.1), which explains the targeted ranking model's behavior. Our method involves training a Partial Relevance Model (PaRM) as an intermediate step to build a relevance thesaurus. The relevance thesaurus is constructed by scoring candidate term pairs using the trained PaRM model. It is evaluated extrinsically by how well it correlates with the targeted neural model's predictions when used to complement vocabulary mismatch in traditional IR systems.

Through manual inspection of the relevance thesaurus, we identify two key findings about the behavior of neural ranking models that are trained on MS MARCO: (1) the *postfix-a* finding, which reveals that the models treat the character "a" appended to a term as equivalent to a quotation mark due to encoding errors in the training data; (2) the *car-brand bias*, which suggests that the models exhibit biases towards certain car brands when ranking documents; (3) the *temporal bias*, which indicates that the models consider years in the distant future or past to be more strongly associated with the query term "when"

compared to the current year and its immediate vicinity. Our experiments using multiple state-of-the-art neural information retrieval models demonstrate that such behaviors are not only found in the cross encoder that we mainly targeted but also replicated in multiple IR models, highlighting the potential value of our explanation method.

The contributions of this work are as follows:

C10 We propose that a relevance thesaurus can be used as a global explanation for neural relevance models.

C11 We introduce a novel method for building a relevance thesaurus, which leverages our novel architecture Partial Relevance Model (PaRM) as an intermediate representation.

C12 We provide qualitative insights about the unexpected behaviors of the models, highlighting underlying data issues and apparent data induced biases in the model.

CHAPTER 2

RELATED WORK

The related works are grouped into three parts.

In section 2.1, we describe the background for chapter 3 and chapter 4. It covers history of the natural language inference (NLI) task, followed by applications and explanations for NLI. As we propose explainable NLI models in these chapters, we introduce previous efforts on explainable (NLI) models and how our works differ and/or improve over them.

In section 2.2, we describe background for chapter 5. Here, we summarize the works on machine learning model explanations, with focus on evaluations for explanations, which serve as foundation for our proposed evaluation metrics.

Finally, in section 2.3, we provide background for chapter 6. We discuss global model explanations in other machine learning tasks, and explain why they are not applicable for explaining information retrieval tasks. Then, we contrast our task of global explanations from existing explanations for IR models or more interpretable IR models.

2.1 Explain NLI with classification role labeling

The natural language inference (NLI) task aims to classify the logical relationship (entailment, contradiction, or neutral) between a given premise and hypothesis pair. In our work, we target NLI models that can be trained from the Multi-Genre NLI Corpus (MNLI) (Williams et al., 2018). MNLI is the most frequently used NLI dataset due to its large size (400,000 sentence pairs) and covering multiple genres.

2.1.1 NLI Models

Adoption of an attention mechanism in neural networks has contributed to the improvement of the NLI systems. One notable example is the Decomposable Attention Model, which utilizes Long Short-Term Memory (LSTM) networks to encode input tokens and subsequently combines the encoded representations using an attention mechanism (Parikh et al., 2016). Compared to later models, this model features a single attention layer, which simplifies the process of inferring alignments between the input tokens.

Since the introduction of the Transformer architecture (Vaswani et al., 2017) in 2017, it has become the dominant architecture for Natural Language Inference (NLI) tasks. As explained earlier, the Transformer architecture consists of multiple transformer blocks, each containing multi-head self-attention and feed-forward layers.

The use of contextualized embeddings from pre-trained language models has become essential for NLI models (Devlin et al., 2019a; Yang et al., 2019). In recent years, advances in NLI performance on benchmarks such as the Multi-Genre Natural Language Inference (MNLI) dataset have been primarily driven by improvements in pre-trained language models (Raffel et al., 2019; Radford et al., 2019).

2.1.2 Explain NLI

Camburu et al. (2018) built the **e-SNLI** dataset on top of a well-known dataset SNLI (Bowman et al., 2015) by adding textual explanations for each instance (Table 2.1). It also contains token-level annotations that represent the important tokens for the decision. They reported that it was challenging to create quality explanations and evaluate generated sentences, because it is not easy to come up with clear criteria to define a good explanation. They proposed to train a neural network that generates explanation sentences. Using e-SNLI, Thorne et al. (2019) have investigated whether the attention component of the neural network can be used to generate token-level explanations. The results were not positive: they showed that the explanation score derived from the attention score is less effective

Premise: An adult dressed in black holds a stick. Hypothesis: An adult is walking away, empty-handed. Label: contradiction Explanation: Holds a stick implies using hands so it is not empty-handed.
Premise: A child in a yellow plastic safety swing is laughing as a dark-haired woman in pink and coral pants stands behind her. Hypothesis: A young mother is playing with her daughter in a swing. Label: neutral Explanation: Child does not imply daughter and woman does not imply mother.
Premise: A man in an orange vest leans over a pickup truck. Hypothesis: A man is touching a truck. Label: entailment Explanation: Man leans over a pickup truck implies that he is touching it.

Table 2.1: Examples from e-SNLI. Annotators were given the premise, hypothesis, and label. They highlighted the words that they considered essential for the label and provided the explanations (Camburu et al., 2018).

than the generic machine learning explanation method LIME (Ribeiro et al., 2016b). In chapter 3, we show our approach is more precise than LIME and the attention component based approach.

There is work on building alignment rationale for NLI (Jiang et al., 2021). They suggested compactness, contiguity and fidelity as three important factors for good alignment explanation. These objectives are more focused on readability and model-grounded evaluations, which is different from our direction. Moreover, it is limited in differentiating the roles of the tokens. The alignments alone do not indicate whether the paired tokens are entailed, contradicting or neutral tokens. However, our task definition in chapter 3 requires these to be differentiated into different tags (match, conflict and mismatch).

2.1.3 Interpretable models

Due to the limited interpretability of Transformer-based models, there were efforts to build an NLI model which is inherently interpretable. Wu et al. (2021) proposed the Explainable Phrasal Reasoning (EPR) model. EPR builds phrase-level inferences through the pipeline composed of three components: chunking, alignment, and classification. Chunk-

ing is driven by manual rules based on syntactic features. The chunks are mostly composed of noun-phrases and verbal-phrases, frequently excluding functional words. To align premise chunks to hypothesis chunks, chunks are encoded by Sentence-BERT encoder (Reimers and Gurevych, 2019), and the most similar chunks are paired. The classification prediction involves initially constructing local decisions, which are then combined to form the global decision (final text-pair classification). Local decisions are built on each of the aligned chunk pairs. The local decision is combined by static formulas motivated from fuzzy logic.

Our work in chapter 4 is influenced by the EPR model, and adopts its formulas to build global decisions from local decisions. Our model is largely different in the chunking parts. Our model is trained to work on variable segmentation choices, while EPR only works in fixed segmentation based on syntactically grouped phrases. By combining different segmentation choices, our model can build precise attributions to tokens while maintaining higher task accuracy.

Levy et al. (2013) proposed an idea of partial textual entailment and adapted existing textual entailment models for the tasks. This work looks on the surface similar to our work in chapter 4 but they solve different technical challenges. First, the models they used were not neural models and explicitly built features by comparing tokens from two texts in a bag of word style. Second, they used a facet, which is a tuple of words, as the granularity that the partial entailment decisions are made. Thus, our work in chapter 4 solves a new challenge that cannot be answered by this work.

Krishna et al. (2022) proposed a ProoFVer model, which generates a sequence of inferences steps which determines the classification decisions. The model takes a claim sentence (corresponds to hypothesis) and an evidence sentence (corresponds to premise) and generates triplets where each triplet is composed of a span from the claim, a span from the evidence and a natural logic operator. The generated evidence span is supposed to be an aligned span for the claim span, and the natural logic operator indicates the relation be-

tween the two spans. The training data is heuristically generated based on the sentence level annotations. The sentences are segmented (chunking) and aligned by a neural chunker (Akbik et al., 2019) and aligner (Sabet et al., 2020) which are not directly supervised or designed for the task. Then, the appropriate natural logic operators for the given span pairs are searched with manual rules, lexical resources such as WordNet (Miller, 1995) and manual annotations for popular patterns and exceptional cases.

2.1.4 Neural network explanation methods

Various methods have been proposed to explain the predictions of neural networks, each with its own advantages and limitations. Certain neural explanation methods, called explanation generators, incorporate additional machine learning components beyond the original models and have their own parameters to be trained (Gilpin et al., 2018). As mentioned earlier, a number of these approaches have used human-written explanations as supervised labels (Hendricks et al., 2016; Camburu et al., 2018; Huk Park et al., 2018). While supervised approaches can be effective and easy to implement when the target data is accompanied by explanations, such as textual descriptions of images (Hendricks et al., 2016), it is less common for text data to have additional explanatory text. Furthermore, as discussed in the introduction, the criteria for token-level rationale annotations can be more ambiguous and costly than the text pair-level annotations. Thus, the methods to infer token-level rationales from text-pair level signals are desirable.

Certain neural explanation methods, called explanation generators, incorporate additional machine learning components beyond the original models and have their own parameters to be trained (Gilpin et al., 2018). As mentioned earlier, a number of these approaches have used human-written explanations as supervised labels (Hendricks et al., 2016; Camburu et al., 2018; Huk Park et al., 2018). While supervised approaches can be effective and easy to implement when the target data is accompanied by explanations, such as textual

descriptions of images (Hendricks et al., 2016), it is less common for text data to have additional explanations annotated.

In contrast to explanation generators, other neural network explanation approaches aim to be more generic and unsupervised. These approaches can be broadly categorized into gradient-driven methods and perturbation-based methods.

Gradient-driven methods assign an importance or salience score to the input by examining the current gradients or the currently active weights, which are mostly decided by activation of non-linearity units. Initially, the gradient from the input to output was used as the importance score (Simonyan et al., 2014). Later approaches used complex combinations of the gradient at multiple points (Sundararajan et al., 2017). The layerwise relevance propagation (LRP) method explains the contribution of the input tokens by recursively distributing the contribution of an upper layer's neurons to the lower layer's neurons based on the weights at the particular input instance (Bach et al., 2015). This method has been employed to explain a number of text classification problems – for example, sentiment analysis using recurrent neural network (Arras et al., 2017b) and document classification using convolutional neural networks (Arras et al., 2017a). It shows which input words are important for a particular word-generation or classification decision.

One potential pitfall of gradient-driven methods is that they may not be reliable outside the small faithful locality. Many methods only examine the gradients (or weights) at single inputs, which makes it challenging to capture a larger view. For example, we found that the impact of negation such as "not" is often underestimated, and yet its deletion may change the classification decision from entailment to contradiction. Our approach and other explanation generation approaches are more robust in handling this problem compared to gradient-driven methods, as they are trained to generate larger locality during training.

Perturbation-based approaches change part of the input and examine the changes to the output and the network (Zeiler and Fergus, 2014; Sundararajan et al., 2017). One easy way to arrive at a larger view of neural network behavior is to use these methods (Du et al.,

2018). However, a major drawback of perturbation-based approaches is computational cost. Moreover, the effect of removing multiple tokens simultaneously might be very different from the effect if they are removed independently. In addition to the cost of executing exponential permutation candidates, translating the permutation behavior into a localization decision is not trivial.

2.1.5 Novelty of our work

Our work in chapter 3 and chapter 4 aims at identifying token-level entailment or contradiction information with explicit criteria for three different token-level labels.

In contrast, most of generic model explanation works focus on differentiating relative importance between the tokens. Thus, they do not clearly differentiate roles of each token. For example, in contradicting pair, it does not differentiate if a token is important because it is showing contradictory aspects (e.g.,"empty" in Table 2.1), or because it is indicating common aspects (e.g.,"hand" in Table 2.1)

Our work differs from supervised approaches in sequence tagging tasks, where a model is trained with the token-level annotations and tested in the same distribution, because we aim at extracting token-level information from the text-pair classification tasks. We expect that the semantic understanding in text-pair classification tasks such as entailment or relevance is more diverse and potentially applicable to broader applications than token-level annotations which are only defined and collected for narrow domains and formats.

2.2 Model explanations and Alignment Rationales

Many machine learning model explanation approaches aim at assigning feature importance parts of the inputs (Simonyan et al., 2014; Ribeiro et al., 2016b; Li et al., 2016; Sundararajan et al., 2017). In natural language processing tasks whose inputs are texts, the term explanations or rationales are often represented as real valued scores of the input tokens or a selected subset of the tokens (Lei et al., 2016; Li et al., 2016).

2.2.1 Evaluating model explanations

In the model explanation literature, sufficiency and necessity metrics are often used to measure faithfulness of explanations (DeYoung et al., 2020; Carton et al., 2020). Sufficiency measures if the given explanation is sufficient to result in the original decision. Necessity (also called comprehensiveness) measures if the explanation covers all the important evidence and is measured by removing the explanation part of the inputs and checking if the model decision changes. These two metrics do not penalize rationales for being too verbose and thus favor longer rationales. When explanation models provide real-valued scores for input tokens, the rationales can be forced to be concise by selecting the top k% of the tokens as the final rationales (Jiang et al., 2021), but this strategy is not applicable when the explanation models only provide binary decisions so there is no ordering. In our work, we propose a modification of the *necessity* metric to control verbosity, which is applicable even when tokens are given only binary scores.

2.2.2 Attention-driven model explanations

Extracting an alignment rationale for natural language inference was studied by Jiang et al. (2021). The alignment is built and evaluated based on the attention mask. Specifically, the attention vector across two segments is removed if they are not in the alignment. There are two notable limitations of this work. First, the method and evaluation is dependent on the specific architecture of the model. Second, in the case of BERT-based models, the attention flow inside the same segment and the flow to special tokens ([CLS] or [SEP]) are always kept. Thus, the alignment could be built through these tokens even when the direct attention vectors are dropped.

There are studies to understand the behavior of BERT- or transformer-based models by inspecting their attention weights (Qiao et al., 2019; Zhan et al., 2020). While the supposedly aligned tokens tend to have higher weights than the others, many of the weights might actually not change the model decision when removed (Qiao et al., 2019). Moreover,

there are hundreds of different attention weights between any single token pairs, and how to combine them is yet an unsolved challenge.

2.2.3 Model explanations in information retrieval

Approaches for explaining information retrieval models can be categorized into two groups. The first category relies on interpretable features such as exact match features (Singh and Anand, 2018; Sen et al., 2020; Singh et al., 2021). The second category only selects (or assigns importance to) tokens of the documents as explanation and does not build explicit alignments between each of query terms and the selected document tokens (Singh and Anand, 2019; Verma and Ganguly, 2019; Fernando et al., 2019; Zhuang et al., 2021; Rahimi et al., 2021). Neither category of the existing explanation models are directly applicable for building relevance alignments because no notion of relations between the tokens is considered.

2.3 Relevance thesaurus as global explanations

2.3.1 Global model explanations

Large portions of works on global explanations are for classification tasks on tabular features (Craven and Shavlik, 1995; Boz, 2002; Guidotti et al., 2018), which are suitable for representing numeric or categorical variables such as age or gender. These explanation methods cannot be applied to explain the Transformer architecture which models the relevance between query and document texts.

In NLP tasks, there are works on global explanations for single text classification, which attribute output labels to some words or phrases (Rajagopal et al., 2021; Han et al., 2020). If these methods are naively applied, they could generate explanations that indicate some frequent terms (e.g., "about") are globally important for relevance. Such explanations are not meaningful for IR tasks, where document terms' importance is highly dependent on queries. It would make a more meaningful explanation if it indicates certain terms or

phrases from the query are associated with specific terms or phrases that appear in the document as our work does.

2.3.2 Neural information retrieval and explanations

Existing neural IR models (Dai and Callan, 2019; MacAvaney et al., 2019; Khattab and Zaharia, 2020; Gao et al., 2021; Formal et al., 2021b; Nogueira et al., 2019) are not suitable for modeling context-independent relevance, as they encode a whole sequence with a single Transformer network, allowing attention vectors to flow between them. Thus, the effects of the contexts cannot be isolated and it is not evident whether the contexts are necessary for the identified semantic matching.

While some neural query expansion (Naseri et al., 2021) or document expansion models like doc2query (Nogueira et al., 2019) could seem relevant to our work as they both relate to term matching, the expanded terms are for the whole query or documents and cannot be used to infer the model's behavior in unseen texts.

2.3.3 Traditional information retrieval

Traditional information retrieval methods struggle with semantic matches, where document terms that are not in the query are responsible for signalling relevance (Croft et al., 2010). As the neural models are effectively handling such semantic matches, understanding their semantic match mechanisms would work as a core component in explaining the neural models.

There are works to incorporate semantic matches within the bag-of-word framework. The translation language model for IR (Berger and Lafferty, 1999) is a promising candidate. This model views a query term as a translation of document terms, computing scores for a query term based on the sum of translation probabilities from the document terms. However, their effectiveness was somewhat limited, possibly due to reliance on term co-occurrences statistics (Jing and Croft, 1994; Xu and Croft, 2000), or pseudo-relevance feedback (PRF) for identifying semantically matching terms. To enhance traditional mod-

els with recent advances, Boytsov and Kolter (2021) proposed fine-tuning BERT (Devlin et al., 2019b) for the translation language model (Berger and Lafferty, 1999). This approach, however, is limited to the semantic matches between terms in BERT's subword vocabulary, and does not extend to terms formed from multiple subwords. Moreover, the work lacks analysis or evaluation regarding the explanation perspectives, and does not provide qualitative insights from the outcomes.

2.3.4 Challenge in identifying biases in NLP models

Most works on identifying biases (Zhao et al., 2018) or ensuring model fairness across protected attributes (Coston et al., 2019) presuppose that these attributes, like gender or nationality, are predetermined. They are frequently analyzed through a limited set of manually curated keywords (May et al., 2019). Due to these limitations, if biases exist in terms or concepts beyond what researchers can expect, they become difficult to detect. Our work can address these limitations by representing existing associations through a relevance thesaurus, enabling researchers to identify if any association is inappropriate. In our work, we did not intentionally look for biases related to known protected classes; rather, we came across them by chance when examining a constructed relevance thesaurus (chapter 6).

CHAPTER 3

SEQUENCE LABELING AS EXPLANATION FOR NATURAL LANGUAGE INFERENCE

In this chapter, we introduce our work that aims at solving classification role labeling for natural language inference (NLI). In this work, classification role labeling (CRL) is considered as an "explanation" for natural language inference. This project was published in the ACM Transactions on Information Systems (TOIS) with the title "Explaining Textual Matching on Natural Language Inference" (Kim et al., 2020).

3.1 Task definition

Many efforts in explaining machine learning models in the NLP domain are focused on selecting tokens (parts of the input) that are important for the model's decision. Similar to the previous efforts, we formalize a model explanation for NLI in terms of token-selection (or token tagging). Our work is different in that we propose specific definitions for token-level sequence labeling, which is based on ideal semantic understanding rather than the "importance to the decision", which is not appropriate for crowd-sourced annotations.

The original natural language inference task is a sentence pair classification problem. Two sentences, a premise and a hypothesis, are given. The goal of the task is to classify their relationship into either entailment, neutral, or contradiction (Bowman et al., 2015). We define our goal as a sequence-tagging problem. To provide a clear definition for the token-level annotation, we defined three tags, each of which indicates the role of the tokens in the sentences with regard to the inference decision. Given a pair of input sentences, our goal is to compute a score for each token in the sentences based on how relevant it is to each tag: match, mismatch, or conflict.

Match. The **match** tag in the hypothesis denotes a token whose meaning can be inferred from the premise. A token in the premise is tagged as match if it is required to infer the meaning that appears in the hypothesis. A sentence pair that is labeled as an entailment implies that all the meanings that the tokens in the hypothesis imply can be inferred from tokens in the premise. Thus, we expect all tokens in the hypothesis to be tagged as match and some of the tokens in the premise—those that correspond to the tokens in the hypothesis—to be tagged as match. As a result, many tokens are tagged as match including ones that are trivially the same across the sentences. In the first example in Table 3.1, the hypothesis "Julius made a groaning sound" is entailed by "Julius groaned." Thus, these two parts are annotated as match. The part "The great thing is to keep calm" of the premise does not contain information that corresponds to information of the hypothesis and is not annotated with match.

Mismatch. A token is tagged **mismatch** if it is in the hypothesis but cannot be inferred from the premise in the extreme being completely unrelated. For example, in Table 3.1, "about sports" in the third row cannot be inferred from the premise, and hence is annotated as mismatch. A neutral relationship can be clearly explained by indicating which tokens are considered mismatched. Mismatch is the opposite of match. In the third example of Table 3.1, the part "I liked talking to you" of the hypothesis can be inferred from the premise while "about sports" cannot. Thus, "about sports" is marked as mismatch.

Table 3.1: Three example sentence pairs from MNLI dataset with the corresponding classification labels (entailment, contradiction and neutral) and token-level tags : conflict (red), match (blue) and mismatch (yellow). In each row, the text on the left corresponds to the premise and the text on the right corresponds to the hypothesis.

Premise	Label	Hypothesis
The great thing is to keep calm." Julius groaned.	entailment	Julius made a groaning sound.
yeah i mean just when uh the they military paid for her education	contradiction	The military didn't pay for her education.
uh-huh well I've enjoyed talking to you.	neutral	I liked talking to you about sports.

Conflict. A token is tagged **conflict** if it is a critical token that renders the corresponding concept untrue. We chose to apply the conflict tag only to critical tokens that produce a contradiction, rather than tagging all the tokens of a contradictory concept. In the annotation process, the annotators were instructed not to include tokens that are trivially identical across sentences. The negations and antonym pairs that are relevant to a contradiction are always included. In the second example of Table 3.1 "paid" of the premise and "didn't pay" of the hypothesis is marked as conflict.

3.2 Data collection for evaluation

We built a small dataset to evaluate the proposed token-level tagging tasks. From the dev-match split of MNLI, we annotated 700 instances for each tag, resulting in a total of 2,100 instances. For each tag, 100 instances were used as a development set, and 600 instances were used as a test set. We call this dataset MNLITag.

3.3 Proposed model

Let $f : x \rightarrow z_c$ be the original classification function implemented by the neural network, where x is a sequence of token IDs that are fed to the network. Another function $g : x \rightarrow y_t$ is added to generate an explanation vector y_t for a tag t, where the number of dimensions for y_t is equal to number of tokens in x. Our goal is to train g, so that the score of $y_{t,i}$ (i-th element of y_t) indicates how likely it is that the corresponding input token x_i should be tagged with the particular tag.

Figure 3.1 shows how the original NLI task (text-pair classification) and token-level explanation are modeled using Transformer architecture. The motivation for this modeling is that the vectors for each token are carrying information matching, thus could benefit by multi-task learning.

For each sentence pair in the training data, we select a weak supervision label for each tag and train the explanation generation with it. First, random perturbations (deleting) are

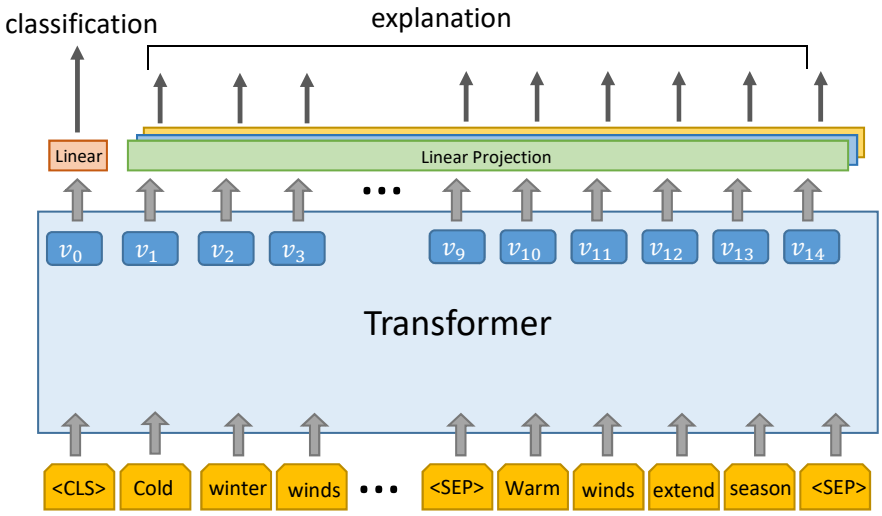


Figure 3.1: The structure of our multi-task model. Thin arrows represent final outputs for classifications and explanations. The red "Linear" in the left represents the linear-projection layer for sentence pair-level classification, and three linear-projection layers on the right are used to generate the explanation scores for each tags. Linear layers with the same color share the same parameter.

applied to the sentence pair to generate several perturbed inputs. The generated inputs and the original input are then fed into the classification network to obtain the classification probability for the perturbed input. For each of the tags, we select the perturbation that resulted in the most *informative* output changes compared to the output from the original input. We use tokens that were modified in the select perturbation as a weak label to train the network. Figure 3.2 shows a high-level overview of the explanation training. In this section, we describe the details of each step of this training.

Perturbations are applied to each of input sentence pairs by the following procedures: (1) a token index j is randomly selected to start the deletion; (2) from a geometric distribution with $p = 0.5$, the length of the sequence to be deleted, $l \sim G(p = 0.5)$, is sampled; and, (3) l tokens from location j to $j + l$ are deleted. The tokens after the deleted tokens are shifted forward and the end of the sequence is filled with padding tokens. Multiple perturbed instances are generated from a single training instance in this way. Here, the

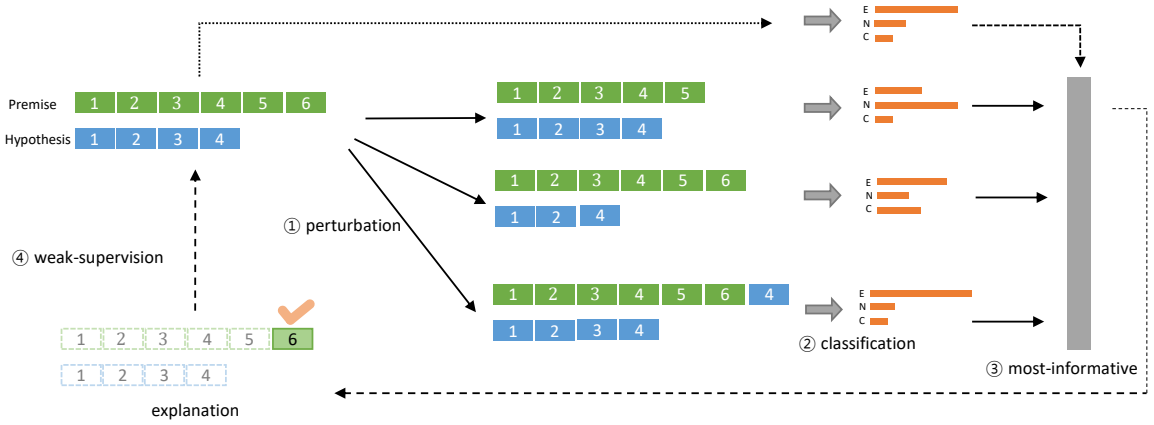


Figure 3.2: Proposed weak-supervision strategy. A number of perturbed instances are generated and they are fed into the classification model. The one that results in the largest output changes is selected as the most informative instance. The model is supervised so that the tokens that are deleted in the most informative instance would have higher scores than the others.

tokens come from a concatenation of the premise and hypothesis, so that both the premise and hypothesis have a chance to be perturbed.

From each input instance x (indices of the sentence pair), a set of perturbed instances $\{x^{(1)}, x^{(2)}, ..., x^{(m)}\}$ is generated. These perturbed instances are fed into the network to produce a corresponding classification probability (softmax) output.

Among the perturbed instances, we want to **select** the one that changes the output the most in a way that we are interested in. We will call such instance as the **most informative instance**. We define and measure the degree of informativeness by what we refer to as a signal function. We expect that modeling each of the tags separately would help to represent the different aspects of the textual understanding (match, mismatch and conflict). For each target tag, we define a corresponding signal function $\mathcal{S}_L$ as follows:

$$\begin{aligned} \mathcal{S}_{match}(x) &= f_e(x) \\ \mathcal{S}_{conflict}(x) &= f_c(x) \\ \mathcal{S}_{mismatch}(x) &= f_n(x), \end{aligned} \tag{3.1}$$

where f_e, f_c, and f_n are softmax probability outputs for entailment, contradiction and neutral, respectively.

Because long sequences of deleted tokens are likely to result in larger output changes that are less meaningful, we penalize any perturbation with a large number of token changes by introducing a size penalty:

$$\mathcal{D}(x, x^{(k)}) = \max\{0.1 \cdot (d - 3), 0\}, \tag{3.2}$$

where d is the number of modified tokens between x and $x^{(k)}$. Thus, if two different perturbations cause similar changes in the signal function, the shorter one would be preferred. The numbers in the penalty term were heuristically designed to match the scale of $\mathcal{S}_t(x) - \mathcal{S}_t(x^{(k)})$ which is in the range [-1, 1].

Equation 3.3 shows the final informative score for each perturbed instance $x^{(k)}$, where $\mathcal{S}_t$ is one of the signal functions in Equation 3.1:

$$\mathcal{I}_t(x, x^{(k)}) = \mathcal{S}_t(x) - \mathcal{S}_t(x^{(k)}) - \mathcal{D}(x, x^{(k)}). \tag{3.3}$$

For each tag t, the most informative instance $\hat{x}$ is selected from the perturbed instances:

$$\hat{x}_{(t)} = \underset{k \in [1,m]}{\arg\max}\, \mathcal{I}_t(x, x^{(k)}), \tag{3.4}$$

where m is the number of perturbed instances. As a result, we obtain three instances, one for each of the tags. For some input instances, it is possible that even the largest change of the signal function (the model's output) is very small. For example, consider the case where a sentence pair is classified as entailment and there is very low probability for contradiction. It is possible that any deletion does not change the contradiction probability much. In this case, the most informative instance for conflict would not be meaningful

enough as the magnitude of change is very small. We handle this problem by setting a minimum threshold on the informative score, so that the instance for the particular tag is rejected and the training is skipped when the most informative perturbation is not of adequate quality. We selected a threshold of 0.3, because this is approximately the average value for the probability of each label (three probabilities that total 1).

Finally, the **weak-supervision** signal is decided by the most informative instance $\hat{x}_{(t)}$. We take the tokens that were modified from x to $\hat{x}_{(t)}$ as the weak label for the tag t. We denote the weak label for tag t as $\hat{y}_t$.

Let y_t be a vector representing the model output where each dimension $y_{t,j}$ is the score for j-th token to be important for the tag t. The size of y_t equals the number of tokens in the input sequence. Given a weak supervision label $\hat{y}_t$, the cross-entropy loss for each tag t is given as

$$\mathcal{L}_t = -\sum_j \hat{y}_{(t),j} \log y_{t,j}, \tag{3.5}$$

where the label $\hat{y_{t,j}}$ is 1 if j-th token was modified in the perturbation and 0 otherwise. Our final loss is sum of the loss for each tag

$$\mathcal{L} = \sum_t \mathcal{L}_t. \tag{3.6}$$

As an alternative to classical cross-entropy function, we suggest using Pearson's correlation coefficient as a loss function. The loss function that we refer to as correlation loss is given as

$$\mathcal{L}_t = -\frac{\sum_j (\hat{y}_{(t),j} - \bar{\hat{y}}_{(t)})(y_{(t),j} - \bar{y_t})}{\sigma_{\hat{y}_t} \sigma_{y_t}}, \tag{3.7}$$

where the label $\hat{y_{t,j}}$is 1 if j-th token was modified and -1 otherwise [1]. σ_{y_t} and $\bar{y_t}$ are the standard deviation and the mean of the values in vector y_t

$$\bar{y_t} = \frac{\sum_j y_{t,j}}{|y_t|} \tag{3.8}$$

$$\sigma_{y_t} = \sqrt{\frac{\sum_j (y_{t,j} - \bar{y_t})^2}{|y_t| - 1}}. \tag{3.9}$$

$|y_t|$ is the size of the explanation vector, which is equal to the maximum sequence length. $\sigma_{\hat{y}_t}$ and $\bar{\hat{y}_t}$ are defined similarly for the vector $\hat{y}_t$. We adopted this loss function as we expect this could be more robust than cross-entropy loss with noisy signal. This correlation loss function satisfies the conditions for an effective list-wise loss function (Xia et al., 2008). The effect of the loss function is discussed in section 3.4.4.

3.4 Experiments

We evaluated our method and the baseline approaches by comparing them against human annotated sequence tagging. Our experiments were mainly conducted on the model trained on the MNLI dataset (Williams et al., 2018), which we annotated based on the definition in section 3.1. For comparison with previous work, we also conducted the experiment on the e-SNLI dataset (Camburu et al., 2018). The annotation definition of the e-SNLI is slightly different from ours, because they did not explicitly define the role of the tokens as match, mismatch or conflict. We observed that most methods were applicable to both the e-SNLI data and our dataset.

3.4.1 Implementation

Recent state-of-the-art models for the NLI tasks are built by fine-tuning a pre-trained language model (Raffel et al., 2019; Liu et al., 2019). We used the pre-trained uncased BERT model with 12 layers and fine-tuned the entire network.

[1] Using 1 and 0 would be effectively the same

We first trained 2.5 epochs only for the NLI classification task. We then began training both the classification and explanation modules. We alternately processed classification training steps and explanation training steps. Both the classification and explanation was trained only on the training split. The explanation training lasted for 0.5 epochs, which is roughly 12,000 steps with a batch size of 16. For each training instance, 20 perturbed inputs were generated, from which the most informative pair were selected. Our training was done with single M40 GPU. The training with explanation took roughly 33 hours to be trained. The training without explanation took 21 hours to be trained.

A parameter update was performed using the Adam optimizer with weight decay. Linear decay of the learning rate and warm-up steps were applied as they are in the original implementation of the BERT model. For the initial learning rate we used $2 \cdot 10^{-5}$. The maximum sequence length was set to 300 tokens.

During the explanation training we also trained the classification module by alternating the two tasks every step. For the explanation training, we used a smaller learning rate than we did for the classification training (0.3 times the learning rate for the classification training).

3.4.2 Evaluation

3.4.2.1 Metrics

The metrics we used in the evaluation were accuracy, mean average precision (MAP), and precision at 1 (P@1). To evaluate accuracy, we tuned the cut-off threshold on the development set to maximize accuracy. Accuracy was measured over all tokens in the test set. MAP and P@1 do not require any cut-off threshold.

3.4.2.2 Data annotation

Conflict was labeled only for sentence pairs whose gold label was contradiction. Similarly, *match* was labeled for the sentence pairs with entailment label and *mismatch* for neutral label.

Because the "entailment" label implies that the content of the hypothesis can be inferred from premise, if the label is "entailment", all tokens in the hypothesis should be labeled *match*. Thus, we evaluated the *match* label only on the premise sentences. For *mismatch*, we evaluated tokens only in the hypothesis.

Forty percent of the data were annotated by three annotators. When the annotators produced different annotations, the annotation that is more similar to the others was selected. Thus if two annotators made similar decisions and the other made a different decision, one of the two similar decisions was selected.

From the validation split of MNLI, we annotated 700 instances for each tag, resulting in a total of 2,100 instances. For each tag, 100 instances were used as a development set, and 600 instances were used as a test set. Kohen's κ for token-level agreement was 0.74.

3.4.2.3 Baselines

To show the characteristics of the dataset with trivial baselines, we included random and inverse document frequency (Idf) approaches. The random method assigns a random score to each token. The Idf method assigns each token a score of $(1/df)$ where df is the number of sentences in the collection that contain the corresponding word. P@1 of the random method is approximately the proportion of true label.

LIME (Ribeiro et al., 2016b) is a generic classifier explanation method that has been shown to be the best-performing method in previous work on the e-SNLI dataset (Thorne et al., 2019). Given an input, the LIME method generates numerous perturbed variations of the input. It evaluates the model's outputs for these variations and builds a linear classifier that can predict the model's output near the given point. This method requires a large number of perturbed instances for each input. For the number of perturbed inputs, we selected the proposed value from the implementation.[2]

[2]https://github.com/marcotcr/lime.

We considered three gradient-driven approaches: Saliency (Simonyan et al., 2014), Grad*Input (Shrikumar et al., 2017), and Integrated Gradient (IntGrad) (Sundararajan et al., 2017). The Saliency method evaluates the score of each input as the absolute value of the input's gradient toward the output. The Grad*Input method obtains the score by multiplying each input dimension by the gradient. The IntGrad method evaluates the score by numeric integration of the gradient over the input changes from starting value to current input value. These three methods were implemented based on the DeepExplain library (Ancona et al., 2018).[3] Modifications were applied to each method to support word embedding.

Saliency, Grad*Input, IntGrad and LIME were designed to generate scores for each dimension of the input. As our explanation is token level, we sum the score for each dimension of the token's embedding. Taking a maximum was also considered, but the results from the development data showed that the maximum is similar to or worse than the sum.

We used two perturbation-based methods. The *Sensitivity* method assigns each token a score according to the change in the output when the token is deleted (Zeiler and Fergus, 2014). *Sensitivity (M)* deletes multiple tokens simultaneously. As it is infeasible to try all possible deletions, this method samples the location and length of the sequence to delete. Each token's score is assigned by the maximum change of outputs among the attempted deletions. For comparison, we allowed an equal number of runs for Sensitivity and Sensitivity (M).

As our model uses sub-word tokens, the scores of sub-word tokens were translated into a token-level score by taking the maximum of each token's sub-word tokens' scores.

3.4.3 Results

We refer to our method as **SE-NLI** (Self-Explaining NLI). In Tables 3.2 and 3.3, SE-NLI (CO) and SE-NLI (CE) denote our methods with different loss functions: Pearson's

[3]https://github.com/marcoancona/DeepExplain.

correlation coefficient (Equation 3.7) and cross-entropy loss (Equation 3.5), respectively. In the remaining parts of this chapter, SE-NLI without any notation refers to SE-NLI (CO).

3.4.3.1 Performance on original NLI task

In this subsection, we demonstrate the performance of our model on the original NLI classification task to show that our multi-task learning for explanation approach does not have negative effect on the performance in the original task. As discussed in the related work (section 2.1), recent improvement in the NLI task has been mostly driven by improved language model pre-training. Thus, newer models are not particularly different from the perspective of the NLI task itself. Following existing work (Xin et al., 2020; Gupta and Durrett, 2019), we include a comparison of models trained from the same BERT$_{BASE}$ checkpoint. Table 3.2 shows the accuracy of the classification-only model and our multi-task trained models on the MNLI dataset, all having the same BERT$_{BASE}$ as a starting point. The models show little difference in the classification.

3.4.3.2 Comparison with alternative explanation methods

Table 3.2: Original NLI task (entailment, contradiction and neutral) accuracy of the models trained with our explanation generator and the model that was only trained for the classification task. All three models used the same BERT$_{base}$ model for parameter initialization. The numbers are accuracy on MNLI-matched split. The accuracy difference between runs 1, 2, and 3 are not significant, showing P-values of 0.60 (1 vs 2), 0.41 (1 vs 3) and 0.19 (2 vs 3).

	Model	Accuracy
1	Classification only	84.4
2	SE-NLI (CO)	84.5
3	SE-NLI (CE)	84.2

Table 3.3 shows the results of the token-level explanation tagging conducted on MNLI. In most cases, SE-NLI (CO) is the best-performing method. The cross entropy version, SE-NLI (CE) is often comparable to the other methods, but it does not perform as well as SE-NLI (CO). It is surprising to find that SE-NLI performs much better than Sensitivity and

Table 3.3: Experiment on token-level tagging done on MNLI. For each column the highest value is marked with bold text. If the highest value is significantly better than all the other methods it is marked with ▲(p = 0.01). Average # of Runs represents the number of neural network required to explain a single instance.

	Conflict			Match			Mismatch			Avg
Method	P@1	MAP	Acc	P@1	MAP	Acc	P@1	MAP	Acc	#Runs
Random	0.289	0.431	0.762	0.593	0.673	0.509	0.537	0.623	0.519	-
Idf	0.364	0.504	0.762	0.703	0.710	0.508	0.478	0.609	0.517	-
Saliency	0.705	**0.733**	0.762	0.813	0.793	0.524	0.798	0.761	0.530	1
Grad*Input	0.426	0.486	0.761	0.737	0.703	0.507	0.598	0.639	0.523	1
IntGrad	0.559	0.582	0.786	0.868	0.744	0.506	0.652	0.689	0.539	300
LIME	0.637	0.618	0.799	0.905	0.777	0.597	0.735	0.731	0.601	5,000
Sensitivity	0.601	0.598	0.780	0.950	0.795	0.590	0.653	0.674	0.542	39.8
Sensitivity (M)	0.398	0.520	0.762	0.658	0.728	0.508	0.723	0.764	0.523	39.8
SE-NLI (CO)	**0.750**▲	0.723	**0.800**	**0.965**	**0.903**▲	**0.760**▲	**0.817**	**0.830**▲	**0.714**▲	1
SE-NLI (CE)	0.551	0.599	0.783	0.932	0.874	0.739	0.803	0.803	0.657	1

Table 3.4: Comparison of our method with the reported best methods on e-SNLI dataset. For Thresholded Attention and LIME, the numbers are as presented in the previous work (Thorne et al., 2019). Our own experiments on LIME on BERT based model showed similar numbers to the previous work on LIME. For the comparison we used the same metric as the previous work.

	Premise			Hypothesis		
	P	R	F1	P	R	F1
Thresholded Attention	0.192	0.262	0.222	0.534	0.630	0.578
LIME (LSTM+GloVe based)	0.656	0.483	0.537	0.570	0.669	0.616
LIME (BERT based)	0.376	1.000	0.547	0.460	0.834	0.593
SE-NLI	0.525	0.726	0.609	0.492	1.000	0.660

Sensitivity (seq), because SE-NLI was trained on signals that are similar to those methods. Note that none of the methods were supervised with the explanation annotation. Each tag shows different levels of difficulty mainly due to the different number of positive labels in a single sentence pair. *Match* has the most positive tokens, which resulted in P@1 and MAP being higher than for the other two tags. *Conflict* is the most difficult of all tags.

Among the other methods, Saliency, Sensitivity, and LIME tend to perform better than the other baselines, but none of them performs exceptionally. It is noteworthy that the

Saliency method is among the highest performing methods; it has the simplest implementation and the lowest computational cost.

Comparison with the previous work (Thorne et al., 2019) on the e-SNLI dataset is shown in Table 3.4. The Thresholded Attention method uses attention weights to generate an explanation. We thresholded all the models to maximize the F1 score. Thus, the differences of precision and recall are the results of threshold selection. The scores for Thresholded Attention are from the model built using LSTM and GloVe embeddings. As this model did not benefit from pre-trained contextualized embeddings, such as BERT, it is not directly comparable to SE-NLI. Instead, LIME could be a baseline for the comparison, as our implementation of LIME on BERT showed similar results to the LIME on an LSTM and GloVe based model. On the e-SNLI dataset, SE-NLI out-performed the LIME method by F1. We would not expect Thresholded Attention to be better than SE-NLI, considering that many studies claimed that attention weight alone is insufficient as an explanation (Jain and Wallace, 2019).

3.4.3.3 Computational requirements

Table 3.3 shows on the right the average number of neural network runs required for each method. Saliency and Grad*Input need to compute one forward run and one backward run (gradients to input) to compute the token-level scores. The IntGrad method uses numeric integration over the multiple points of gradients and outputs, and so it requires a large number of computations: the default parameter from the implementation is 300. The LIME method requires many outputs of perturbed inputs to build a linear classifier: 5,000 is also from the default parameter of the implementation. The Sensitivity method deletes each token in the input one by one, and 39.8 is the average number of tokens in the evaluation data. We did not count forward runs and backward runs as separate runs if they used the same input. Saliency, Grad*Input, and IntGrad require both forward runs and backward runs; the other methods use only forward runs. Along with two other methods, SE-NLI has

the lowest computational requirements, requiring only a single run to generate an explanation. If we assume that computing both forward runs and backward runs is more expensive than only computing forward runs, SE-NLI has the lowest computational requirement of all the methods during the prediction time.

Compared to the other methods, SE-NLI requires additional computation during training. However, the additional computational cost are of reasonable amount. In our implementation, we trained the explanation generator for only 0.5 epochs, whereas the whole training procedure for NLI lasted over 3 epochs.

SE-NLI requires additional computation to get outputs from a number of perturbations. However, these additional computations are still affordable, because forward runs for perturbations are much faster than back-propagation and parameter updates.

3.4.3.4 Effect of loss functions

It is notable that the model trained with cross-entropy loss (SE-NLI (CE)) dramatically fails on *conflict* tags. In the early stage of our experiment, we observed that using the cross-entropy converges slower than the correlation loss. The difference in the final accuracy (precision) between cross-entropy loss and correlation loss was not as significant when we used a much larger learning rate without decaying. However, that configuration had an observable negative effect on the original classification task.

The motivations of using the correlation loss was that cross-entropy loss would penalize the predictions (location in the sequence) that are not in the weak label (most informative instances) more harshly than correlation loss does. However, cross-entropy loss exhibits better accuracy, suggesting that correlation loss is better for ranking metrics.

3.4.4 Analysis

3.4.4.1 Fidelity

We evaluated the fidelity of our explanation with a deletion experiment, which is commonly used in attribution analysis papers (Arras et al., 2017a). We selected 2,000 sentence

pairs whose gold labels were contradiction. We deleted tokens in decreasing order of conflict tag scores and measured how much the average accuracy changed. Figure 3.3 shows that SE-NLI is good at predicting the tokens that will make the system's accuracy plummet much faster when deleted.

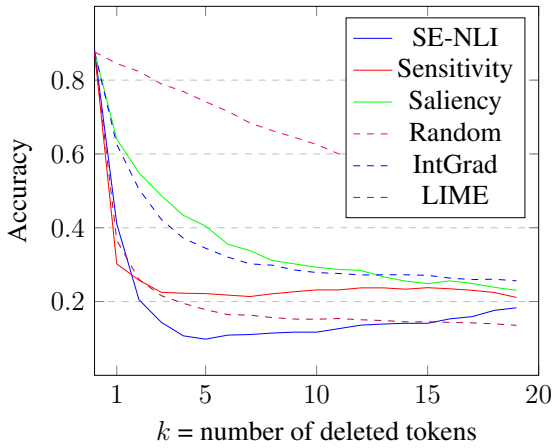


Figure 3.3: Sentence classification accuracy changes as tokens are deleted in the order of decreasing scores of explanation prediction. Rapid accuracy drops are considered evidence of a good explanation (Arras et al., 2017a).

3.4.4.2 Multi-task learning

It is unclear whether NLI knowledge is actually needed for the token-tagging task. To investigate this question, we trained the explanation generator in a separate network. For training data, we recorded weak-supervision input from the training of SE-NLI and applied it to the target network. We tested two cases: in one, the model was initialized with pre-trained BERT (BERT start), whereas in the other, the parameters were randomly initialized (Cold start). Table 3.5 shows the results of alternative models. The BERT start model shows comparable performance for the match tag, but it does not reach the performance of the original model for the other two tags. Thus, we conclude that there is meaningful gain in using multi-task learning for explanation generator.

Table 3.6 shows a case that highlights two models with different levels of language understanding. Although the word "camping" does not carry conflicting meaning, the model

Table 3.5: Effect of multi-task learning. MTL with NLI is the same as the model SE-NLI (CO) in Table 3.3. BERT start and cold start were trained using the same supervision as the training of SE-NLI but they were trained on vanilla BERT or a random initialization rather than on an NLI-trained model.

	Conflict			Match			Mismatch		
Model	P@1	MAP	Acc	P@1	MAP	Acc	P@1	MAP	Acc
MTL with NLI	0.750	0.723	0.800	0.965	0.903	0.760	0.817	0.830	0.714
BERT start	0.625	0.640	0.798	0.965	0.890	0.754	0.783	0.791	0.688
Cold start	0.484	0.544	0.775	0.700	0.711	0.584	0.537	0.628	0.523

Model	Sentences
MTL with NLI	**P:** I don' t know um do you do a lot of camping **H:** I know exactly.
BERT start	**P:** I don' t know um do you do a lot of camping **H:** I know exactly.

Table 3.6: Comparison of conflict prediction of MTL model (MTL with NLI) and baseline model (BERT start). **P** stands for premise and **H** stands for hypothesis.

without NLI knowledge (BERT start) assigns it a high score. Moreover, this model assigns a lower score to the token "know" in the hypothesis than it does to the "know" token in the premise.

3.4.4.3 Hyper-parameters

In this subsection, we demonstrate the change in the model's performance as the hyper-parameter values changes.

When the tokens in the inputs are deleted for perturbations, the number of deleted tokens for each perturbation is sampled from a geometric distribution. We found that deleting a flexible number of tokens is superior to deleting only one token. Figure 3.4 shows that the MAP changes as this parameter changes. The score decreases if too many tokens (0.9) or too few tokens are deleted. The value of p being 0 implies that always single token is deleted, and, in this setting, MAP score for the *match* tag drops. We expect that the score

drop is much larger on the match tag because it requires that a greater number of tokens be tagged. Specifically, for the match tag, 63% of the tokens in a sentence pair are tagged, whereas for conflict, only 30% are tagged. We expect that only deleting a single token causes the model to generate only a few "most important" tokens, which are insufficient to select 60% of tokens for the *match* tag.

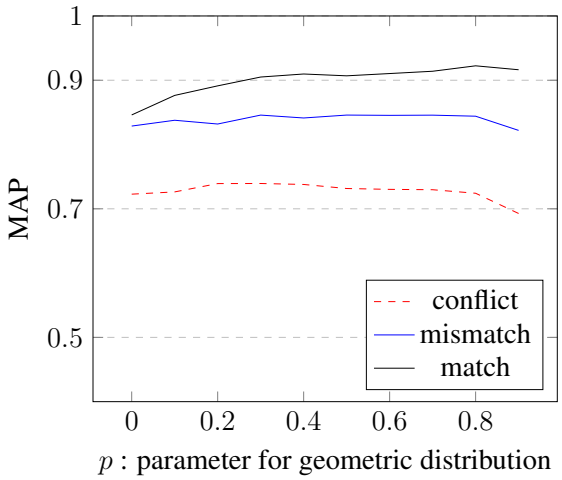


Figure 3.4: Changes in MAP as the parameter p of geometric distribution changes, which decides the length of deleted sequence. Larger p values would result in a longer sequence being deleted.

Another hyper-parameter is the number of perturbations generated when selecting the most informative instance. For the numbers reported above, our method was trained by generating 20 perturbed inputs for each instance. If only a small number of perturbations are considered, even the most informative instance could result in a small difference in outputs. We used the strategy of rejecting the instance and skipping the training when the most informative score is below the threshold (Equation 3.3). This strategy helps the training succeed even when we use small numbers of perturbations, as fewer perturbations lead to more instances having low informative scores. Figure 3.5 shows the MAP scores for the *conflict* tag as the number of perturbations changes. As expected, if the number of perturbations is fewer than five, the accuracy is reduced, and the difference increases when no threshold is applied.

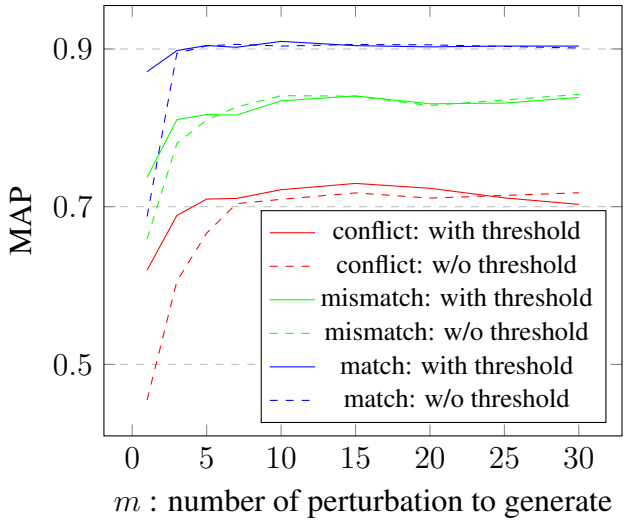


Figure 3.5: Changes in MAP for the *conflict* tag in number of perturbations per step changes.

3.4.4.4 Qualitative analysis

We examined the proposed method's actual output to understand the model's behavior. We considered 18 instances. The NLI task has three labels, so the confusion matrix for the prediction has $3 \times 3 = 9$ entries. Two examples are presented for each of nine entries. We list the first appearing instances in the dataset that matches the entries of the confusion matrix. All the instances were from the validation (matched) split. Tables 3.7, 3.8 and 3.9 each present six examples. Table 3.7 contains examples whose gold label is "contradiction", Table 3.8 contains examples for "entailment" and Table 3.9 contains examples for "neutral".

Although each of three tags could show complementary information, it is difficult to list all three scores for all token in a simple format. Thus, for each example, we presented the single tag that is most relevant to the model's prediction. *Match* (blue) tag scores are displayed for entailment, *conflict* (red) for contradiction and *mismatch* (green) for neutral. Scores are linearly normalized for presentation by color. As there are negative scores, the tokens with white backgrounds could have negative scores.

In the first example of Table 3.7, the model predicted the label to be entailment, and the gold label is entailment. The token "all" in the hypothesis is not considered to be "match". We can at least expect that the model did not consider "all" to match any of the tokens in the premise, but simply treated it as unimportant token. In the third example of Table 3.7, the model assigns a high score to the token "long". We can expect that the model failed to infer that this expression is contradictory to "has never really let".

Similarly, in the third example of Table 3.8, the model assigns a high mismatch score to the token "cold," implying that it concluded that this token contains information that cannot be inferred from the premise.

3.5 Conclusion

In this chapter, we investigated an approach to generate explanations for NLI.

1. We defined token tags that show the role of the token in three classification decisions (entailment, neutral and mismatch).

2. We described a new weak-supervision training method for an explanation generator.

3. We proposed a neural model that contains both an explanation-generating function and a sentence classification function in a shared network.

4. We showed that our proposed model outperforms strong baselines, while our model has the least computational cost of those considered.

The proposed method exhibits certain limitations that suggest potential areas for future work.

Since the weak supervision is driven from perturbation, it inevitably inherits the limitations of perturbation-based explanations. For example, some perturbations could largely change the interpretation of the texts by removing contexts and would result in unreasonable explanations. MNLI is more robust to perturbations as it contains many informal and

Prediction (Label)	Sentences
entailment (contradiction)	**P:** The most important directions are simply up and up leads eventually to the cathedral and fortress commanding the hilltop, and down inevitably leads to one of three gates through the wall to the new town. **H:** Go downwards to one of the gates, all of which will lead you into the cathedral.
entailment (contradiction)	**P:** But uh these guys were actually on the road uh two thousand miles from from home when they had to file their uh their final exams and send them in **H:** These men filed their midterm exams from home.
neutral (contradiction)	**P:** What' s truly striking, though, is that Jobs has never really let this idea go. **H:** Jobs never held onto an idea for long.
neutral (contradiction)	**P:** Even if you' re the kind of traveler who likes to improvise and be adventurous, don' t turn your nose up at the tourist offices. **H:** There' s nothing worth seeing in the tourist offices.
contradiction	**P:** This site includes a list of all award winners and a searchable database of Government Executive articles. **H:** The Government Executive articles housed on the website are not able to be searched.
contradiction	**P:** Yeah i i think my favorite restaurant is always been the one closest you know the closest as long as it' s it meets the minimum criteria you know of good food **H:** My favorite restaurants are always at least a hundred miles away from my house.

Table 3.7: Our model's explanation score output for examples whose gold labels are **contradiction**. Different tags are shown depending on the model's actual prediction. If the model's prediction is entailment, the scores for the *match* tag are highlighted blue. For neutral predictions, the *mismatch* scores are highlighted green. For contradiction predictions, the *conflict* scores are highlighted red. **P** stands for premise, and **H** stands for hypothesis.

Prediction (Label)	Sentences
entailment	**P:** Uh i don' t know i i have mixed emotions about him uh sometimes i like him but at the same times i love to see somebody beat him **H:** I like him for the most part, but would still enjoy seeing someone beat him.
entailment	**P:** You and your friends are not welcome here, said severn. **H:** Severn said the people were not welcome there.
neutral (entailment)	**P:** I' m not sure what the overnight low was **H:** I don' t know how cold it got last night.
neutral (entailment)	**P:** Mortifyingly enough, it is all the difficulty, the laziness, the pathetic formlessness in youth, the round peg in the square hole, the whatever do you want? **H:** Many youth are lazy.
contradiction (entailment)	**P:** And uh as a matter of fact he' s a draft dodger **H:** They dodged the draft, i' ll have you know.
contradiction (entailment)	**P:** I' m kind of familiar with the weather out that way in west Texas but not in not in lewisville **H:** I do not know the weather conditions in lewisville.

Table 3.8: Our model's explanation score output for examples whose gold labels are **entailment**. Different tags are shown depending on the model's actual prediction. If the model's prediction is entailment, the scores for the *match* tag are highlighted blue. For neutral predictions, the *mismatch* scores are highlighted green. For contradiction predictions, the *conflict* scores are highlighted red. **P** stands for premise, and **H** stands for hypothesis.

Prediction (Label)	Sentences
entailment (neutral)	**P:** Tuppence rose. **H:** Tuppence floated into the air.
entailment (neutral)	**P:** What changed? **H:** What was unique?
neutral	**P:** The new rights are nice enough **H:** Everyone really likes the newest benefits
neutral	**P:** Calcutta seems to be the only other production center having any pretensions to artistic creativity at all, but ironically you' re actually more likely to see the works of satyajit Ray or mrinal Sen shown in Europe or North America than in India itself. **H:** Most of mrinal Sen' s work can be found in European collections.
contradiction (neutral)	**P:** Um- hum um- hum yeah well uh i can see you know it' s it' s it ' s it' s kind of funny because we it seems like we loan money you know we money with strings attached and if the Government changes and the country that we loan the money to um i can see why the might have a different attitude towards paying it back it' s a lot us that you know we don' t really loan money to to countries we loan money to governments and it' s the **H:** We don' t loan a lot of money.
contradiction (neutral)	**P:** I' m not opposed to it but when its when the time is right it will probably just kind of happen you know **H:** I cannot wait for it to happen.

Table 3.9: Our model's explanation score output for examples whose gold labels are **neutral**. Different tags are shown depending on the model's actual prediction. If the model's prediction is entailment, the scores for the *match* tag are highlighted blue. For neutral predictions, the *mismatch* scores are highlighted green. For contradiction predictions, the *conflict* scores are highlighted red. **P** stands for premise, and **H** stands for hypothesis.

ungrammatical texts. In a task where being ungrammatical affects a model decision, it may be more problematic. In chapter 4, we will propose a method which better handles this limitation. chapter 5 will also illustrates such limitations of perturbations.

CHAPTER 4

CONDITIONAL NATURAL LANGUAGE INFERENCE USING CLASSIFICATION ROLE LABELING

In this chapter, we further investigate the sequence tagging task, classification role labeling (CRL) which was introduced in chapter 3. We target the scenario where we want to explain apparently contradictory claims and we formalize it as the task of identifying tokens that indicates opposite outcomes and different conditions, which we name Conditional-NLI (Cond-NLI). Cond-NLI is similar to CRL on Natural Language Inference (NLI), as both requires identifying tokens that could cause the sentence pairs to be classified as entailment, neutral or contradiction. We observed that existing perturbation-based CRL methods suffer in this task, which lead us to propose a new model. This part is published in the Findings of ACL: EMNLP 2023, with the title "Conditional Natural Language Inference" (Kim et al., 2023).

The Cond-NLI extends traditional Natural Language Inference (NLI) to better capture a full spectrum of information in textual relationships. Traditional NLI involves determining whether a given premise entails, contradicts, or remains neutral to a hypothesis, typically through a three-way classification model.

The need for Cond-NLI becomes evident when we consider real-world applications, particularly in the biomedical domain. To illustrate, Table 4.1 presents two claims from biomedical articles (Dahlöf et al., 2002) and (Matsui et al., 2008), included in the Potentially Contradictory Claims (PCC) corpus (Alamri and Stevenson, 2016). At first glance, these claims appear contradictory, but a closer look will show that they address different conditions and treatments. Therefore, they should not be classified as contradiction but

Question: In patients with advanced diabetes, does treatment with antihypertensives improve renal function or protect against cardiovascular incidents? **Claim1:** Interpretation Losartan prevents more cardiovascular morbidity and death than atenolol for a similar reduction in blood pressure and is better tolerated. [Ans: **Yes**] **Claim2:** Although a bedtime dose of doxazosin can significantly lower the blood pressure, it can also increase left ventricular diameter, thus increasing the risk of congestive heart failure. [Ans: **No**]

Table 4.1: An example from the BioClaim dataset. Tokens in red indicate opposite outcomes (contradiction), and yellow ones indicate different conditions (neutral).

neutral in the traditional NLI framework. However, this classification poses a significant challenge. Labeling such claims as neutral can obscure the nuanced differences between truly unrelated claims and those like our biomedical examples, which are related but not contradictory due to specific contextual factors. Thus, the Cond-NLI task aims to refine the understanding of these relationships, moving beyond the limitations of traditional NLI. This refinement is crucial for efficiently mining large sets of neutral-labeled claims, allowing for a more comprehensive and nuanced understanding of a given question or topic.

We develop a modeling framework to capture the relationship between a pair of sentences that provides different answers under diverse conditions. Such sentence pairs are henceforth referred to as *conditionally-compatible*, since none of the *entailment*, *contradiction*, or *neutral* classes of NLI precisely describes their relationship.

Cond-NLI includes two token-level tasks – one is to identify contradictory tokens that embody contradictory aspects and the second is to identify neutral tokens that indicate conditions that are not entailed by the other sentence. The focus of this study is to determine different conditions in a pair of conditionally-compatible sentences. For the example pair in Table 4.1, the segments highlighted in yellow represent the condition tokens. Contrary to NLI, where an ordering is specified between paired sentences via the roles of premise and hypothesis, paired sentences in Cond-NLI do not require such an order because the contradiction holds in both directions.

Automatic identification of different conditions in conditionally-compatible sentence pairs allows us to summarize and provide a full spectrum of answers in a form where users are not overloaded with excessive information. This is of particular practical importance as it has shown that there are usually multiple answers to a user's question in different domains, such as biomedical (Alamri and Stevenson, 2016), e-commerce (Santos et al., 2011), and factoid question-answering (Min et al., 2020), where the difference between answers/opinions is their provided conditions.

We propose Partial-ATtention model, PAT, a simple yet effective model for natural language inference that can address the Cond-NLI task. PAT predicts an NLI label for a sentence pair from the intermediate labels for their partitions. The intermediate labels for partitions of sentences can be subsequently used to attribute these labels into the token-level.

The NLI token-level attributions align closely with the objective of Cond-NLI . Different conditions in a claim pair would cause an NLI model to predict the pair to be neutral. Thus, identifying the tokens responsible for triggering neutral labels could serve as a technique to detect different condition tokens in Cond-NLI. Similarly, contradictory tokens of Cond-NLI can be attained from attributing contradiction label in NLI. Finally, PAT effectively solves Cond-NLI through training with sentence-level NLI data, without requiring task-specific token-level annotations.

To evaluate different models for Cond-NLI, we build (and make publicly available) the BioClaim dataset, an extension of an existing corpus initially built to assist systematic reviews (Alamri and Stevenson, 2016). The BioClaim dataset provides a challenging benchmark for the NLI models. In contrast to the SciEntsBank dataset (Dzikovska et al., 2013), which lacks contradictory sentence pairs, BioClaim includes conditionally-compatible sentence pairs. Such pairs require the identification of neutral tokens in the presence of contradictory tokens. Compared to other token-level explanation datasets such as e-SNLI and MNLITag (Camburu et al., 2018; Kim et al., 2020), which are built on NLI corpora (Bow-

man et al., 2015; Williams et al., 2018), BioClaim has longer hypothesis sentences. This characteristic introduces additional complexity in the selection of non-entailed tokens.

Perturbation-based methods (Ribeiro et al., 2016b; Kim et al., 2020) have shown to be effective in identifying tokens that contribute to contradiction or neutral labels when evaluated on e-SNLI (Camburu et al., 2018) or MNLITag (Kim et al., 2020). However, we show that these perturbation-based explanation models face challenges in accurately identifying condition tokens when hypothesis sentences are long and contain a large number of non-entailed tokens (conditions in Cond-NLI).

Extensive experiments on the BioClaim and SciEntsBank (Dzikovska et al., 2013) datasets show that our PAT significantly outperforms strong and state-of-the-art baseline models. Against IntructGPT (Ouyang et al., 2022) and ChatGPT (OpenAI, 2022), PAT shows better performance on the SciEntsBank dataset and comparable performance on the BioClaim dataset, while PAT has a significantly smaller number of parameters. While our PAT model slightly underperforms the cross-encoder BERT model on the original NLI task, its enhanced interpretability enables effective fine-grained token-level inference required for Cond-NLI.

4.1 Cond-NLI task and datasets

4.1.1 Task definition

Our Conditional Natural Language Inference (Cond-NLI) is formally defined as a token-level classification task, aligning with the definition of the existing task of partial entailment (Levy et al., 2013). Given a pair of claims (p, h) and a span s from h, the goal is to classify s as either neutral or contradictory to p. Note that, neutral tokens are considered equivalent to condition tokens.

4.1.2 BioClaim

To evaluate our model, we built the BioClaim dataset by adding token-level annotations to an existing corpus of potentially contradictory claims (PCC) (Alamri and Stevenson, 2016). PCC consists of 24 closed-form research questions and a total of 259 claims relevant to the questions. The claims are aligned with the relevant questions and are also annotated with their answer (Yes or No) to the relevant questions. Claim pairs relevant to the same question with different answers to the question are potentially contradictory or conditionally-compatible.

From 24 question groups, we selected pairs with opposite answers (Yes-No). Since each group has different numbers of Yes or No labeled claims, the combinations of opposite-answer pairs range from 3 to several hundred. We limit the maximum number of pairs from each group to 20, prioritizing those with greater term overlap when sampling.

Annotators were given a sampled claim pair and asked to annotate tokens that indicate opposite outcomes (corresponding to the contradiction label) and tokens that indicate different conditions in the two claims (corresponding to the neutral label). While NLI has three classes, we only annotated tokens that are related to contradiction and neutral, as the entailment tokens are expected to be the remaining tokens that are not contradiction nor neutral.

We employed nursing college students as annotators. The resulting dataset consists of 14,915 annotated tokens, including 1,862 contradiction tokens and 6,145 neutral tokens, all of which are derived from 285 claim pairs. Using Cohen's Kappa (Cohen, 1960), we observed a moderate agreement score of 0.46. Out of all the claim pairs, 195 received multiple annotations; we randomly selected two annotations from these pairs to measure agreement.

In the evaluation of Cond-NLI using BioClaim, each claim pair generates multiple Cond-NLI problems. This occurs for every token in the claim pair (tokenized by spaces) and for each token-level class, namely neutral and contradiction.

4.1.3 SciEntsBank

We also used SciEntsBank (Dzikovska et al., 2012), a dataset with fine-grained entailment annotations, for our evaluation due to its task similarity with neutral token classification in Cond-NLI. SciEntsBank was built to assess student answers, and formatted as an entailment task by taking a student answer as a premise and a reference answer as hypothesis. It annotated if a facet of the hypothesis is entailed by the premise, where a facet is a tuples consisting of two words. Following a data filtering process similar to one used in SemEval-2013 (Dzikovska et al., 2013), the test split contained 9,974 'Expressed' and 10,516 'Unaddressed' facet-level annotations.

4.2 Partial-Attention NLI Model

The typical effective approach for text-pair classification, such as the NLI task, using Transformer-based language models such as BERT (Devlin et al., 2019a), is by concatenating the text pair as input, which we refer to as *full cross-encoder* BERT. Specifically, cross-encoder BERT takes the concatenation of premise p and hypothesis h, denoted by $p \circ h$, taking the [CLS] token vector as sentence representation, and outputs classification probability $\mathbf{y}$ as:

$$\mathbf{y} = f(p \circ h). \tag{4.1}$$

Output $\mathbf{y}$ in the NLI task is a 3-dimensional vector representing the probabilities of the entailment, neutral, and contradiction classes.

We propose the Partial-ATtention model, PAT, that predicts the NLI label for p and h based on two intermediate NLI labels for two subsequences of h. Specifically, the hypothesis h is partitioned into two subsequences h_1 and h_2. Premise p is separately concatenated with h_1 and h_2 and fed into the encoder f', which outputs intermediate predictions ρ_1 and ρ_2, respectively. Each intermediate output is a probability distribution over three classes.

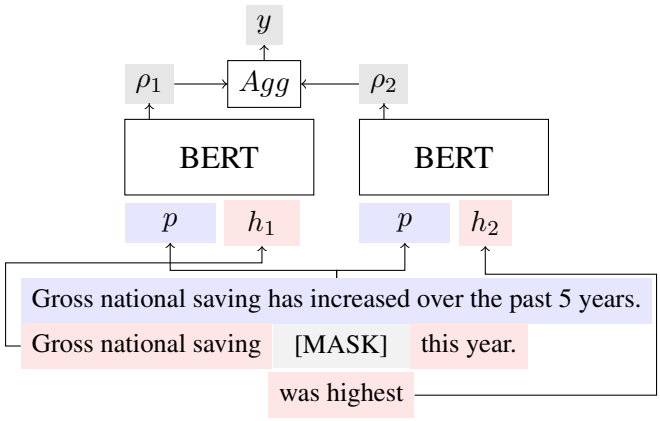


Figure 4.1: The architecture of proposed PAT model. p represents the tokens of the premise. h_1 and h_2 are subsets of hypothesis tokens. Agg combines two intermediate output ρ_1 and ρ_2 as in Eq. 4.5.

The intermediate NLI labels are then aggregated to obtain the final NLI label for the pair p and h:

$$g(p, h) = \text{Agg}(\rho_1, \rho_2) \tag{4.2}$$

$$\rho_1 = f'(p \circ h_1), \quad \rho_2 = f'(p \circ h_2), \tag{4.3}$$

where f' has the same architecture as the function f in Eq. 4.1, but is trained to be robust to partial text segments. The function $\text{Agg}(.)$ combines the intermediate outputs to predict the final NLI label. Figure 4.1 shows the PAT architecture.

Partitioning hypothesis. For training, h is partitioned by randomly selecting two indices i_s and i_e, where $i_s \leq i_e$. h_1 is built from tokens i_s to i_e of h. h_2 is built by concatenating two segments of h with a [MASK] token between them: token 1 to $i_s - 1$ and token $i_e + 1$ to the last token of h.

Combining intermediate decisions. The expected logical behavior of the aggregation function, when each intermediate decision is discrete, is shown in Table 4.2. For example, when both intermediate decisions, ρ_1 and ρ_2, are entailment (the probabilities for entailment are close to 1), the final decision y should be entailment (entailment probability is close to

		ρ_1		
		Entailment	Neutral	Contradict
ρ_2	Entailment	Entailment	Neutral	Contradict
	Neutral	Neutral	Neutral	Contradict
	Contradict	Contradict	Contradict	Contradict

Table 4.2: Logical behavior for combining the intermediate NLI decisions. Gray cells show the final NLI label.

1). If one of the two intermediate decisions, for instance ρ_1, is neutral (contradiction) while the other is entailment, then the combined decision inherits the label of ρ_1. If one is neutral and the other is contradiction, the final decision should be contradiction. This is similar to the methods proposed by Wu et al. (2021) and Stacey et al. (2022), which are motivated by fuzzy logic.

To implement this logical behavior, we first model Table 4.2 with an integer matrix

$$M = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 1 & 2 \\ 2 & 2 & 2 \end{bmatrix}, \tag{4.4}$$

where entailment, neutral and contradiction are represented as 0, 1 and 2, respectively. Based on this matrix, we then build a one-hot representation T. T is a rank 3 tensor where $T_{ijk} = 1$ if $M_{ij} = k$ and $T_{ijk} = 0$ otherwise. The final NLI label $\mathbf{y}$ is obtained by the matrix multiplication:

$$\mathrm{Agg}(\rho_1, \rho_2) = \rho_1^T \cdot T \cdot \rho_2. \tag{4.5}$$

Cond-NLI. Once the PAT model is trained, the intermediate decision predictor f' can be used to predict labels for any arbitrary subsequence s within a hypothesis, as it would treat $p \circ s$ similarly to either $p \circ h_1$ or $p \circ h_2$.

While our goal is to predict a label for an individual token of h, only feeding one token to the model is not ideal due to lack of contextual information. Instead, we consider longer

	MultiNLI	SNLI	SciTail
Cross Encoder	0.829	0.887	0.925
PAT	0.793	0.870	0.889
+ fuzzy logic	0.763	0.844	0.860
+ four segments	0.744	0.831	0.818

Table 4.3: Classification accuracy of the cross-encoder baseline, proposed PAT, and alternative architectures (ablation study) for sentence-pair NLI.

spans that contain the token in h. The tokens' final label is determined by combining the labels of these spans.

Specifically, we used sliding windows of size 1, 3 and 6 tokens with a stride of 1. Let S_i denote the set of subsequences that contain the i-th token of h. The probability vector c_i indicating three NLI classes of i-th *token* of h with respect to p is predicted as:

$$c_i = \frac{1}{|S_i|} \sum_{s \in S_i} f'(p \circ s), \tag{4.6}$$

where $f'(p \circ s)$ is a probability vector of three classes from the intermediate predictions of PAT.

4.3 Experiments

Experimental Settings. Both the full cross-encoder NLI (Eq. 4.1) and the PAT models are trained by fine-tuning the BERT-base model (Devlin et al., 2019a) on the MultiNLI dataset (Williams et al., 2018) for one epoch, as more epochs are expected to result in overfitting and lower performance on the BioClaim dataset. For perturbations and token-level enumerations, sentences are tokenized by spaces instead of BERT's subword tokenizer.

4.3.1 NLI sentence-pair classification

We compare the accuracy of the full cross-encoder BERT and PAT for the original NLI task over three datasets MultiNLI, SNLI (Bowman et al., 2015), and SciTail (Khot et al., 2018). Both models are separately trained and tested on each of the datasets.

Table 4.3 summarizes the accuracy of the PAT and the full cross-encoder models. PAT shows 2% to 4% lower accuracy than the full cross-encoder model, however intermediate decisions enhance the interpretability of its predicted NLI class.

We also used the accuracy of the NLI task for an ablation study to compare different design aspects of our PAT model, as a higher NLI accuracy is likely to result in good performance for Cond-NLI under similar data distributions. Table 4.3 includes the accuracy of ablated versions of the PAT model. The "+ fuzzy logic" model replaces the our aggregation function in Eq. 4.5 with the one from EPR (Wu et al., 2021), a phrased-based NLI model. The "+ four segments" model, in addition to the previous change, splits the hypothesis into four pieces instead of two in PAT. This is based on the observation that EPR model splits hypothesis into an average of four pieces in the SNLI. We observe that replacing our strategies with those used in the existing models results in lower accuracy over all datasets.

4.3.2 Evaluation Metrics for Cond-NLI

We report accuracy and F1 score as the main metrics for the evaluation of Cond-NLI. For SciEntsBank, we report macro-averaged F1 which is average of F1 scores for each of 'Expressed' and 'Unaddressed' labels (Dzikovska et al., 2013).

Many of the baseline methods such as LIME or SLR, assign (importance) scores to tokens, and do not provide binary class labels. To perform a meaningful comparison that demonstrates the potential of each method, we convert token scores into binary class labels by applying a threshold criterion; tokens are assigned to a specific class depending on whether their scores exceeds the predefined threshold. The threshold is determined through

evaluating multiple candidate values. The chosen threshold for each model is the one that maximizes the model's performance on the validation set.

4.3.3 Baseline methods

We address three research questions in our evaluation. Baseline methods are selected and described based on the research question we aim to address.

RQ1 Is PAT more effective than the lexical match or embedding similarity approaches in classifying neutral/entailed tokens?

Neutral tokens are the ones not entailed by the other sentence in a pair. If a token pair from two sentences has similar meanings (high semantic similarity), one can expect that the tokens are less likely to be neutral. Thus, we consider **exact match** and **word2vec** (Mikolov et al., 2013) as baselines to predict neutral/entailed tokens

A token's entailment score with respect to a sentence is determined by its highest similarity to the other sentence's tokens. For this purpose, we build a similarity matrix $S_{|p|\cdot|h|}$ where S_{ij} indicates the similarity of the i-th token in p to the j-th token in h. In case of exact match, S_{ij} is a binary value indicating whether the two tokens are the same or not. With word2vec, S_{ij} indicates the cosine similarity between embeddings of p_i and h_j. The entailment score of the j-th token in h, h_j, with respect to p is computed as $\max_i S_{ij}$. The neutral score is computed as one minus the entailment score.

In SciEntsBank, a facet s is composed of two tokens of h and we compute the span entailment score as an average of two tokens' entailment score.

RQ2 Is PAT more effective than adapting the existing models for solving Cond-NLI?

First, we investigate if the feature-attribution explanation models (Ribeiro et al., 2016b; Zeiler and Fergus, 2014; Kim et al., 2020) can solve Cond-NLI. These methods assign an importance score to each input feature based on its contribution to the predicted class probability. Given a premise-hypothesis pair and an NLI model, we use feature attribution explainers to obtain importance scores of input tokens to the predicted probability for the

neutral class by the NLI model. Interpreting these importance scores as tokens' neutral scores, feature attribution explainers can solve the Cond-NLI task.

We include the following perturbation-based methods that are either widely-used for explanation of a black-box classifier or specifically designed for explanation of the NLI task. **LIME** (Ribeiro et al., 2016b) is a widely-used explanation method which attributes the model's prediction to input features (tokens in the NLI task). **Occlusion** (Zeiler and Fergus, 2014) removes one token at a time and measures the output changes to score the importance of the removed token. **SE-NLI** (Kim et al., 2020) is an explanation model that generates token-level explanations for the NLI task. It uses BERT token representation as a feature to predict the importance score for each token. The training objective for importance prediction is to predict the change in the NLI scores when the token is deleted.

SLR (Span-Level Reasoning) (Stacey et al., 2022) is an NLI model that makes explicit span-level predictions. However, its span granularity is restricted because it divides hypothesis into spans at noun phrase boundaries. Nevertheless, to demonstrate the limitations of SLR, we converted their span predictions into a token or facet level by using a method similar to Equation 4.6.

Beyond feature-attribution explanation methods, we consider adapting the full cross-encoder NLI model for solving Cond-NLI. The assumption is that if a hypothesis span s is neutral against a premise p, then the NLI model would predict neutral on (p, s), where span s alone is treated as a hypothesis. This baseline can demonstrate the advantage of function f' in Eq. 4.3 over function f in Eq. 4.1. We refer to this baseline as **Token-entail**. Token-entail is different from our PAT model in two ways; it uses the full cross-encoder model in Eq. 4.1 with only a single token as a hypothesis while our model uses sub-sequences of variable length as hypothesis. We did not compare against the full cross-encoder model when the hypothesis is a sub-sequence of longer length, because cross-encoder is not robust to such sub-sequences as input and its performance drops significantly.

We developed the **Co-attention** baseline inspired by the work of Jiang et al. (2021). Co-attention uses the attention scores from a Transformer encoder as a token similarity proxy. The intuition is that in an NLI trained model, a high attention score between a token pair across two sentences indicates that the tokens are likely semantically similar, which makes their representations can be compared through attention. Thus, a token that is neutral is likely to have small attention scores to the tokens of the other sentence. The normalized attention scores of a token to the tokens of the other sentence are averaged over all self-attention heads in all layers. The obtained scores are used as similarity matrix S, similar to the **exact match** baseline.

RQ3 How does PAT compare against GPT-3 based models?

InstructGPT (Ouyang et al., 2022) and **ChatGPT** (OpenAI, 2022), which are fine-tuned versions of the large language model GPT-3 (Brown et al., 2020)), have shown good zero-shot performance in many downstream tasks. To solve Cond-NLI, we used the task instruction used for BioClaim annotation and a claim pair to build a prompt to the LLMs. The LLMs are asked to generate words that correspond to either neutral or contradiction (Figure 4.2) . For SciEntsBank, we included a student answer, a reference answer, and a facet word pair in the prompt (Figure 4.3) and then asked the LLMs to determine if the facet is entailed by the student's answer.

4.3.4 Results

Tables 4.4 and 4.5 show the performance of all compared methods on the Cond-NLI over the BioClaim and SciEntsBank (Dzikovska et al., 2013) datasets. On both datasets, the proposed method, PAT, outperforms other NLI-based methods with the only exception of LIME on contradiction in terms accuracy. However, this gap is not statistically significant and Cond-NLI largely outperforms LIME when evaluated with F1.

We suggest the following reasons for the poor performance of explanation models LIME, Occlusion, and SE-NLI on the Cond-NLI, especially for the neutral class. First,

In each of the examples, two claims extracted from research paper abstracts will be shown. The given two claims seem to be contradictory as they are implying opposite results about the same question. Precisely though, the two claims may have been obtained for different population or intervention details that make it possible that both claims to be true. We want to annotate the tokens (words) that express different conditions.

Claim 1: We conclude that in women with preeclampsia, prolonged dietary supplementation with l-arginine significantly decreased blood pressure through increased endothelial synthesis and/or bioavailability of NO.
Claim 2: Oral L-arginine supplementation did not reduce mean diastolic blood pressure after 2 days of treatment compared with placebo in pre-eclamptic patients with gestational length varying from 28 to 36 weeks.

Condition tokens in Claim 1: women, preeclampsia, prolonged, dietary supplementation, l-arginine, increased, endothelial synthesis, bioavailability, NO

Condition tokens in Claim 2: pre-eclamptic patients, gestational length, 28 to 36 weeks

Figure 4.2: An example of the prompt given to the InstructGPT model to solve Cond-NLI neutral token prediction. The text that is colored with yellow are generated by the model.

Student answer: By letting it sit in a dish for a day.
Reference answer: The water was evaporated, leaving the salt.
Facet: (evaporated, water)

The facet is a relation extracted from the reference answer. In the example above, does the student answer entail the given facet? Answer with Yes/No

Figure 4.3: An example of the prompt given to the ChatGPT model to solve partial entailment task for SciEntsBank dataset.

	Neutral		Contradiction	
	F1	Acc	F1	Acc
Similarity-based				
Exact match	0.647†	0.538‡	-	-
word2vec	0.645†	0.575‡	-	-
NLI-based				
Co-attention	0.644‡	0.538‡	-	-
LIME	0.639‡	0.538‡	0.277‡	**0.872**
Occlusion	0.632‡	0.538‡	0.246‡	0.859‡
SENLI	0.632‡	0.541‡	0.292‡	0.866
SLR	0.624‡	0.538‡	0.280‡	0.859‡
Token-entail	0.638‡	0.538‡	0.248‡	0.866‡
PAT	**0.657**	0.622	0.414	0.871
Large language model				
InstructGPT	0.593‡	0.673‡	0.435	0.856‡
ChatGPT	0.624‡	**0.657**‡	**0.459**	0.846‡

Table 4.4: Cond-NLI: neutral token and contradiction token classification results on BioClaim. ‡ and † indicate that the difference between the method and PAT is significant at $p < 0.01$ and $p < 0.05$.

the hypothesis contains many tokens that are not entailed. Perturbing a small number of tokens is likely to lead to the partial removal of neutral tokens. Such perturbations would cause negligible changes in model predictions. Simultaneously removing all neutral tokens is also unlikely to have a desirable impact on the model decision as large removal increases the chance of out-of-distribution inputs and thus unreliable model decision for explanation (Hase et al., 2021).

Second, many of conditionally-compatible pairs are predicted as contradictory by the NLI models despite the existence of tokens that indicate different conditions. In this case, identifying different conditions becomes more challenging as the neutral probability predicted by the NLI model is very small and effect of not-entailed tokens for the neutral probability cannot be observed.

SLR (Stacey et al., 2022) also underperformed PAT due to its fixed span segmentation, limiting its ability to infer entailment information for arbitrary tokens. The performance of

	UA	UD	UQ	Mean
Similarity-based				
Exact match	0.733	**0.792**	0.753‡	0.759
word2vec	0.753	0.780	0.756‡	0.763
NLI-based				
Co-attention	0.746	0.700‡	0.817	0.754
LIME	0.635‡	0.673‡	0.663‡	0.657
Occlusion	0.494‡	0.488‡	0.404‡	0.462
SENLI	0.542‡	0.547‡	0.600‡	0.563
SLR	0.722‡	0.713‡	0.698‡	0.711
Token-entail	0.714‡	0.721‡	0.713‡	0.716
PAT	**0.763**	0.778	**0.826**	**0.789**
Large language model				
ChatGPT	0.655‡	0.687‡	0.680‡	0.674

Table 4.5: Macro-averaged F1 score on the partial entailment dataset SciEntsBank. UA (Unseen Answers), UD (Unseen Domain), and UQ (Unseen Question) are splits of the test set. ‡ indicates that the difference between the method and PAT is significant at $p < 0.01$.

the token-entail method, which is based on the full cross-encoder, is not as good as PAT. We further inspected its outputs and found that the token-entail method predicts high neutral scores for functional and generic words, such as 'patient', 'study', and 'factors', that are implicitly entailed. These failure examples imply that the full cross-encoder model is not robust to partial hypothesis segments and cannot provide meaningful predictions for them.

Exact match and word2vec outperform other NLI-based methods for predicting neutral tokens in terms of F1 scores on BioClaim and SciEntsBank. However, they cannot be used to predict contradicting tokens, thus their performance for contradiction is not listed. They outperform PAT in Unseen Domain (UD) split of SciEntsBank.

On BioClaim, PAT shows comparable performance to InstructGPT and ChatGPT, since the superiority between them varies depending on the metrics and token classes. Note that GPT-3 has 175 billion parameters (Brown et al., 2020), which is more than 1,000 times larger than our proposed model having 110 million parameters (BERT-base). On SciEntsBank, ChatGPT is not effective, possibly due to the difficulty in connecting a word

pair (facet) to the student answer and reference answer. This format might not be frequent in the data that ChatGPT was trained on.

In BioClaim, the improvements of PAT over all other methods are statistically significant at p-value of 0.01, except for similarity-based methods based on F1, where the significance level is at 0.05. Note that none of the NLI-based method outperformed PAT with statistically significance. The statistically significance was measure by the paired t-test for accuracy and bootstrapping test for the F1 score.

4.3.5 e-SNLI and MNLIEx

We also evaluate our PAT on e-SNLI (Camburu et al., 2018) and MNLIEx (Kim et al., 2020), two token-level annotated datasets, to evaluate its robustness. Although these datasets lack conditionally-compatible sentence pairs, limiting their use for comparing models on the Cond-NLI task, they measure the robustness of PAT across diverse datasets. For MNLIEx, we used the models trained on MultiNLI and for e-SNLI, we used the models trained on SNLI.

Table 4.6 and 4.7 show the performance of PAT and baseline models on token-level explanation datasets e-SNLI and MNLIEx. For this evaluation, we use the metrics and categories that are used in the previous works (Thorne et al., 2019; Kim et al., 2020). Perturbation-based explanation models, LIME and SE-NLI, achieve high performance on these two datasets. The results demonstrate that our PAT does not significantly underperform the explanation model SE-NLI that is designed for and trained on the NLI datasets.

4.4 Conclusion

We proposed PAT, a partial attention model, capable of attributing the model decision into the parts of input. Using PAT, we address the Cond-NLI task, a token-level prediction task that explains conditionally-compatible claims. We built the BioClaim dataset for

Method	Conflict			Match			Mismatch		
	P@1	MAP	Acc	P@1	MAP	Acc	P@1	MAP	Acc
LIME	0.637	0.618	0.799	0.905	0.777	0.597	0.735	0.731	0.601
SE-NLI	0.750	0.723	0.800	0.965	0.903	0.760	0.817	0.830	0.714
Token Entail	0.662	0.628	0.757	0.930	0.842	0.692	0.723	0.733	0.597
PAT	0.696	0.700	0.770	0.918	0.868	0.753	0.850	0.851	0.682

Table 4.6: Token prediction evaluated on MNLIEx (Kim et al., 2020) It show precision at 1 (P@1), mean average precision (MAP), accuracy (Acc).

Method	Premise			Hypothesis		
	Precision	Recall	F1	Precision	Recall	F1
LIME	0.376	1	0.547	0.46	0.834	0.593
SE-NLI	0.525	0.726	0.609	0.492	1	0.66
Token-entail	0.422	1.000	0.560	0.515	1.000	0.649
PAT	0.443	0.939	0.562	0.562	0.959	0.664

Table 4.7: Token prediction evaluated on e-SNLI (Camburu et al., 2018) It show precision, recall, F1 on each of premise and hypothesis. All three labels are averaged without differentiation.

Cond-NLI . The proposed method shows the accuracy up to 8% higher than the best NLI-based baseline method in predicting condition tokens.

We will further demonstrates the effectiveness of PATin chapter 6, where PATis used to identify term-pair level relevance features for information retrieval tasks.

CHAPTER 5

ALIGNMENT RATIONALE FOR QUERY-DOCUMENT RELEVANCE

In this chapter, we investigate the task of building alignment rationales that best explain a query-document relevance classifier. In particular we focus on how we can evaluate the given alignment rationale, as this evaluation plays a crucial role in later optimizing alignment building methods. This part is published as a conference paper in SIGIR 2022, "Alignment Rationale for Query-Document Relevance" (Kim et al., 2022).

BERT-based neural network models have shown state-of-the-art performance in information retrieval tasks (Dai and Callan, 2019; Yates et al., 2021; Craswell et al., 2020). However, due to their complex architectures, they have remained a black box and their underlying decision-making mechanisms are not clear, even to domain experts. There have been efforts to explain black-box models' behavior in terms of the input features (e.g., tokens in document ranking), either by assigning importance scores to the features or selecting a subset of features that are important to preserve the models decisions (Singh and Anand, 2018; Hase et al., 2021; Fernando et al., 2019; Kim et al., 2020)

However, we found few works that answer the alignment question: "If certain document tokens are important for relevance to the query, which part of the query do they respond to?" Figure 5.1 illustrates the goal of alignment. When exact match or soft match based ranking models were used, the alignment between query tokens and document tokens could be acquired with little additional effort. Such alignment information has also used to provide more information to users, such as summarizing and visualizing each of query terms appearances in long document (Hearst, 1995; Hoeber and Yang, 2006), also demonstrating the important of this alignment issue.

Query:	Where is SIGIR 2022
Document:	SIGIR 2022 will be held in Madrid

Table 5.1: An example alignment for the query span 'Where is'

Acquiring alignment has two approaches: (1) aiming at building (ideally) 'correct' or useful alignments regardless of query-document scoring model, or (2) seeking an alignment that best explains (is faithful to) the model. We target the second approach here.

We investigate the possible uses of input perturbation approaches, which make no assumption about the model's internal architecture. If the model outputs different decisions for a perturbed instance (a small change to the inputs), we can expect that the changed features are somehow responsible for the model decisions. To expand feature importance to alignment explanation, one can test if importance of some document tokens depends on the existence of certain query terms. Unfortunately such complex perturbation is more likely to bring undesired consequences such as making the input text ungrammatical (Hase et al., 2021) or changing its meaning drastically such that the model's decision changes more than we would expect given small perturbations. For example, consider the case when the query is "Where is SIGIR 2022" and we want to test which parts of a document are responsible for each of "Where is" and "SIGIR 2022". If we remove "SIGIR 2022" from the query, the query becomes "Where is?". In the case of the BERT-based model trained on the MS-MARCO dataset (Nguyen et al., 2016b), the relevant documents for this reduced query are the ones that contain information about how the expression "Where is?" is used, instead of those that present a location of events or entities.

How often does that happen? Does that actually make the perturbation useless? Are there any fixes if it does? This study addresses these research questions.

The contributions of this chapter are as followings:

1. We propose perturbation-based metrics to evaluate alignment rationale for query-document relevance.[1]

2. We investigate the behavior of the proposed metrics and demonstrate that they are mostly not strong enough to make binary decisions on alignment quality (good or bad), but they can be used to rank two alignment models.

3. We propose that building perturbed instances that are more comparable to the instance being explained, is the key to improvement of evaluation metrics. We showed that a simple approach to get more comparable instances increases the metric coverage from 13% to 68%. [2]

5.1 Alignment Rationales

Let f be a black-box classifier model that given a query q and document d, returns the probability of d being relevant to q, i.e., $f(q, d) \rightarrow [0, 1]$. We assume that the model predicts a document d as relevant to the query q if its output is higher than a pre-defined threshold θ_r, i.e., $f(q, d) \geq \theta_r$, and otherwise predicts it as non-relevant. We use $R(q, d) = 1$ to denote that document d is considered as relevant by the model f, and $R(q, d) = 0$ to denote the non-relevance prediction.

The prediction of model f for a given pair (q, d) can be explained in different formats depending on the desired goal for explanations. We focus on the explanation of text matching between the query and document as text matching has been shown to be a strong signal of relevance.

Assume that q and d are split into two sets of text spans $\mathcal{Q}$ and $\mathcal{D}$, respectively. The segmentation unit can be tokens, phrases, or sentences, and can vary for the query and doc-

[1]Code for reproducing experiments is available at `https://github.com/youngwoo-umass/alignment_rationale`

[2]Based on binary-necessity category.

ument. We consider that each text span of the query indicates one requirement of relevance. Intuitively, the model checks if each of the requirements is satisfied by checking spans in $\mathcal{D}$. Assuming that the model performs such matching process, *alignment* explanations provides more sensible description of model behavior compared to token- or word-level explanations (Ribeiro et al., 2016a).

To evaluate alignment rationales for relevance ranking, we need metrics that capture the degree to which the rationales extracted by an explanation model are in fact contributed to the model prediction. Our goal is to define metrics for evaluating the *faithfulness* of alignment rationales. Once the evaluation metrics are established, they can be used as bases for alignment generation methods, by optimizing the proposed quality metrics via black-box optimization (Hase et al., 2021) or gradient-based methods (Jiang et al., 2021).

Problem Definition. Assume an alignment (qt, dt), where $qt \in \mathcal{Q}$ and $dt \in \mathcal{D}$, is given by an explanation model when $f(q, d) \geq \theta_r$, i.e., the document d is predicted to be relevant to the query q by the model f. The goal is to measure the faithfulness of this alignment to the behavior of model f.

5.2 Evaluation Metrics

We use the two criteria sufficiency and necessity (Carton et al., 2020) in our metrics. **Sufficiency** measures whether a rationale is sufficient for a model prediction by comparing the model output for the full input to its output for the input built from the rationale. **Necessity** measures whether a rationale captures only the necessary information by comparing the model output when the rationale is removed.

We first introduce how these metrics can be used to check alignment-independent rationale for document relevance, and show why they are not suitable for evaluating the faithfulness of alignment rationale explanations. We then propose a new set of metrics.

5.2.1 Alignment-Independent Metrics

Given that $R(q, d) = 1$, a document span dt provides sufficient relevant information if $R(q, dt) = 1$, where the input dt to the model means the document content except the span dt is deleted or masked. Formally, this metric can be defined based on real-valued output (continuous score) or based on binary relevance labels of the model as follows.

$$\text{AI.Suff}(q, d, dt) = -[f(q, d) - f(q, dt)], \tag{5.1}$$

$$\text{AI.Suff}_\text{b}(q, d, dt) = \mathbb{1}[\theta_r \leq f(q, dt)], \tag{5.2}$$

$\mathbb{1}[.]$ is an indicator function, returning a value of one when its condition is satisfied. We use $\cdot_b$ (such as AI.Suff_b) to denote the metrics based on binary outputs. Similar notations are used for the following metrics too. The negative sign is added to make a higher value (closer to zero) of AI.Suff indicate a higher quality of rationale.

The sufficiency metric prefers longer spans of documents as explanations. For example, in the extreme case of selecting the entire document as an explanation, the metric will have the highest value. To address this issue, we propose a modification of the *necessity* metric (Carton et al., 2020) for relevance ranking. Let $d \setminus dt$ denotes a text acquired by removing the span dt from document d. Our *necessity* metric considers the span dt as having only the necessary relevant information and being compact if $R(q, d \setminus \hat{dt}) = 0$ for all non-empty $\hat{dt} \subseteq dt$. This metric penalizes long explanations containing non-relevant information.

$$\text{AI.Ness}(q, d, dt) = f(q, d) - \text{avg}_{\hat{dt} \subseteq dt} f(q, d \setminus \hat{dt}) \tag{5.3}$$

$$\text{AI.Ness}_\text{b}(q, d, dt) = \mathbb{1}[f(q, d \setminus \hat{dt}) < \theta_r] \tag{5.4}$$

It is computationally expensive to compute the outputs of a deep neural model for several subsets of each candidate span. Therefore, we randomly sampled 10%, 20%, ..., 100% of dt as $\hat{dt}$ and averaged the model predictions for these subsets.

While these metrics evaluate whether span dt has contributed to the model's relevance prediction, they do not evaluate whether it has been aligned with query span qt or not. These definitions are all based on deletion perturbations of the instance (q, d) to be explained. We thus start by extending these metrics for evaluation of the alignment faithfulness using deletion perturbations.

5.2.2 Deletion-based Metrics

To evaluate alignment rationales, we consider simultaneous perturbations of the query and document in the instance to be explained. One would intuitively expect that if a document is relevant to a query, it is also relevant to any span of the query. Specifically, when $R(q, d) = 1$, the expectation is to get $R(qt, d) = 1$. If this assumption is satisfied by the model, we can perturb the document to extract the span dt that affects prediction $R(qt, d) = 1$ and validate the influence of alignment (qt, dt) in the model prediction $f(q, d)$. Given the condition $R(qt, d) = 1$, the evaluation metrics are then formally defined as follows.

Sufficiency. Span dt provides sufficient relevant information for span qt if $R(qt, dt) = 1$. This metric is referred to as $\mathrm{D.Suff}$.

$$\mathrm{D.Suff}(q, d, qt, dt) = -[f(qt, d) - f(qt, dt)] \tag{5.5}$$

$$\mathrm{D.Suff_b}(q, d, qt, dt) = \mathbb{1}[\theta_r \leq f(qt, dt)] \tag{5.6}$$

Necessity. Span dt contains only the necessary relevant information for span qt if $R(qt, d \setminus \hat{dt}) = 0$ for all non-empty $\hat{dt} \subseteq dt$.

$$\mathrm{D.Ness}(q, d, qt, dt) = f(qt, d) - f(qt, d \setminus \hat{dt}) \tag{5.7}$$

$$\mathrm{D.Ness_b}(q, d, qt, dt) = \mathbb{1}[f(qt, d \setminus \hat{dt}) < \theta_r] \tag{5.8}$$

5.2.3 Substitution-based Metrics

Deletion-based metrics rely on the implicit assumption that $R(qt, d) = 1$ when $R(q, d) = 1$. However, this assumption frequently fails. For example, the ranker (Dai and Callan, 2019) that is used in our experiments predicted that the document in Figure 5.1 is relevant to the query "Where is SIGIR 2022", but it is not relevant to the query "Where is". To address this issue, we propose to substitute the query parts other than qt instead of deleting them.

We introduce a new query $qt \cup w$, which is built by substituting spans of $q \setminus qt$ with spans w. For the example query "Where is SIGIR 2022", qt can be "Where is". Deletion-based metrics use the model prediction for query "Where is" to compute faithfulness. Instead, we substitute "SIGIR 2022" with $w =$"CIKM 2022", and probe the model with the new query "Where is CIKM 2022". As spans w are newly introduced to query, it is likely that the document does not contain any information about w. Thus, we also add w to span dt of the document so that the w part of the new query has exact match in the document. Substitution allows to more accurately measure if the qt part of the query is satisfied by the document span dt.

Sufficiency. Span dt provides sufficient relevant information for span qt if $f(qt \cup w, dt \cup w) = 1$.

$$\begin{aligned} \text{S.Suff}(q, d, qt, dt) &= -[f(qt \cup w, d \cup w) - f(qt \cup w, dt \cup w)] \\ \text{S.Suff}_{\text{b}}(q, d, qt, dt) &= \mathbb{1}[\theta_r \leq f(qt \cup w, dt \cup w)] \end{aligned} \tag{5.9}$$

Necessity. Span dt contains only the necessary relevant information for span qt if $R(qt + w, dt \setminus \hat{dt} \cup w) = 0$ for all non-empty $\hat{dt} \subseteq dt$.

$$\begin{aligned} \text{S.Ness}(q, d, qt, dt) &= f(qt \cup w, d) - f(qt \cup w, dt \setminus \hat{dt} \cup w) \\ \text{S.Ness}_{\text{b}}(q, d, qt, dt) &= \mathbb{1}[\exists w \text{ s.t. } f(qt \cup w, dt \setminus \hat{dt} \cup w) < \theta_r] \end{aligned} \tag{5.10}$$

Table 5.2: Preferences and accuracy of different metrics on two alignment methods: exact match (EM) and random. 'D' in a metric name is for deletion, and 'S' is for substitution. The cases where the difference between the deletion and substitution metrics are statistically significant ($p < 0.01$) are denoted with *. The numbers in bold (substitutions) are the ones that we consider better compared to the corresponding deletion based version.

Metrics			Relative preference			Accuracy	
			EM	Random	Equal	EM	Random
Continuous	Attention Mask		0.50	0.50	0.00	0.86	0.86
Binary	Necessity	$\mathrm{D.Ness_b}$	0.13*	0.00*	0.87*	0.98*	0.87*
		$\mathrm{S.Ness_b}$	0.66*	0.02*	**0.32***	0.92*	0.33*
	Sufficiency	$\mathrm{D.Suff_b}$	0.78*	0.00	0.22*	0.83*	0.06*
		$\mathrm{S.Suff_b}$	0.81*	0.01	**0.18***	0.97*	0.16*
Continuous	Necessity	$\mathrm{D.Ness}$	0.85	0.15	0.00	0.99*	0.87*
		$\mathrm{S.Ness}$	0.86	0.14	0.00	0.94*	0.34*
	Sufficiency	$\mathrm{D.Suff}$	0.97	0.03	0.00	0.83*	0.06*
		$\mathrm{S.Suff}$	0.97	0.03	0.00	0.97*	0.16*

Substitution candidates. When substitution spans have the same syntactic role and similar semantic category as $q \setminus qt$, the new query is more comparable to the original query. However, we found that even without such complex selection of w, $qt \cup w$ can provide more reliable estimate of model behavior. To get substitution candidates, we first collect all term-level n-grams of the target retrieval collection for values of n ranging from 1 to 4. A span w from the obtained n-grams will be used for the computation of the substitution-based metrics if $f(qt \cup w, w) = 0$. This condition allows to prune the large candidate space and to make sure that selected spans are not specific enough that their matching alone is enough for relevance prediction by the model for $(qt \cup w, d \cup w)$. If no span w satisfies the condition, the lower bound score is assigned to $\mathrm{S.Suff}$ and $\mathrm{S.Ness}$.

5.3 Experiments

The experiments demonstrate how the metrics proposed in Section 5.2 are different. We are especially interested in comparing deletion-based versus substitution-based metrics and binary versus continuous metrics.

Table 5.3: Accuracy of exact match alignments for different units of query-side targets (qt).

		word	low-idf spans	high-idf spans
Attention Mask		0.86	0.62	0.95
Necessity	$\mathrm{D.Ness_b}$	0.98	0.88	0.64
	$\mathrm{S.Ness_b}$	0.92	0.90	0.73
Sufficiency	$\mathrm{D.Suff_b}$	0.83	0.21	0.74
	$\mathrm{S.Suff_b}$	0.97	0.75	0.91

Dataset. We use the BERT-based document ranker as our target function f to be explained (Dai and Callan, 2019). We trained the model with MSMARCO document ranking dataset (Craswell et al., 2020), and perform alignment evaluation on the dev split. The ranker is trained with the cross-entropy loss, thus can be considered as a binary classifier. The trained ranker showed NDCG@10 of 0.625 on TREC Deep Learning Track 2019 (Craswell et al., 2020), which matches the performance reported by the similar models (Craswell et al., 2020).

Our main evaluation set consists of 3,176 cases where each case consists of a unique triple (query, text, query-side target qt). These cases are obtained by selecting 50 queries and the documents that are predicted to be relevant to them. We split documents by sentences, and filtered sentences that are predicted to be relevant to the query when they are fed individually. In the main evaluation setting, individual words are used as a query-side target.

Alignments. We analyze the behavior of each metric on two alignment methods: exact and random matches. The exact match alignment is built by selecting any word in the document that overlaps with the words of the query-side target qt. The overlap is compared in sub-word level. Random alignments are built by randomly selecting document tokens. Random alignments are controlled to have the same number of tokens as the exact-match alignments.

We assume that the exact match alignments are better than random *on average*. Thus, we can expect that an ideal metric prefers exact-match over random alignments. This does

not imply that the ideal metric should prefer exact-match over random for every case since it is possible that in some cases random alignments may be better than the exact-match alignments. When measuring relative preferences of alignments by evaluation metrics, the cases where no exact match exists are excluded.

We also compare our metrics with another evaluation metric for alignments based on attention masks (Jiang et al., 2021). This metric drops attention flows between the two segments (query and document), except the token pairs that are predicted to be aligned. Section 2.2.2 provides more details about this metric. For the attention-mask metric, the binary version is not applicable because changing the attention mask results in a change of the model score by a small magnitude only, which does not flip the classification label (always relevant). Following Jiang et al. (Jiang et al., 2021), the absolute difference of logistic scores are used to compute the metric.

Results. Table 5.2 shows the results of the various metrics on the two types of alignments. Relative preference shows how often exact-match or random alignment is preferred over the other by a metric. We removed the cases where exact match (EM) and Random had the same alignment prediction. Thus, the equal column of the table indicates the rates that randomly-aligned and exact-match tokens get the same preference by an evaluation metric, while the tokens are different. Accuracy indicates the rate that the score given by a metric is over the $\theta_r = 0.5$ (in case of attention mask, lower than θ_r).

First, we observe that the attention-mask metric does not behave as expected. It prefers random alignments in almost half of the cases, which implies that this metric is capturing something different than the alignment rationales for relevance ranking. We investigated these cases to find out which tokens appear when the random alignment is preferred. The tokens for some special characters such as "." or "?" appear more often in the preferred cases than their average frequencies. This implies that if an alignment contains "." or "?", it is more likely to be preferred over the exact match by the attention-mask metric compared to when the alignment contains other random tokens. One potential reason can be that the

BERT-based ranker is using the tokens for these special characters to combine information, thus the matching tokens (such as common words in the query and document) are compared via these tokens. Another possibility can be that removing attentions between these tokens breaks the score calculation even if they do not play a role in the matching process.

Second, we observe that the `substitution`-based metrics have a lower rate of equal decisions compared to their corresponding `deletion`-based metrics. We conclude that the high equal rate of necessity-`deletion` metric indicates a clear failure of the metric. First, the cases with the same alignment were removed, thus compared alignments are always different. Second, the dataset is known to have many exact match terms between the queries and documents, thus a certain portion of exact-match alignments should be considered better than the random ones. Thus, we conclude that the `substitution`-based metrics have advantages over the `deletion`-based metrics. We approximate the "coverage" of a metric as the portion of the data that the metric makes two different decisions on two different alignments. In case of binary-necessity, the deletion-based metric has coverage of 13% (13% + 0%) and substitution-based metric has coverage of 68% (66% + 2%).

We believe that the high accuracy of the necessity metrics is probably resulted from the perturbed queries (qt or $qt + w$) not being comparable to their corresponding original queries, thus yielding non-relevant predictions for all perturbations.

Next, we compare the binary metrics against their continuous versions. With the continuous metrics, the equal rate decreased to near zero. A large portion of the equal cases by the binary metrics are classified as preference to exact matches by the continuous metrics. From this trend, we expect that continuous metrics, that are sensitive to small differences in model scores, could be capable of preferring better alignments. Reduction in equal cases of the `Necessity`metric is mostly observed for cases that $f(qt, d)$ is near zero, $f(qt, d \setminus \hat{dt})$ for exact match dt is also near zero and is lower than $f(qt, d \setminus \hat{dt})$ for randomly aligned dt.

Query-side target Finally, we compare the evaluation metrics when different segmentation units of queries are used for explanation, i.e., different query targets qt. We built two datasets "high-idf spans" and "low-idf spans". The original dataset consisting of individual words as qt is called "word". For each query, we identified the query terms whose idf (inverse document frequency) values exceed a predefined threshold value. We select a continuous span of the query that covers these high-idf terms. These high-idf spans compose the "high-idf spans" dataset. The remaining low-idf terms, which can be at most two continuous segments per query, compose "low-idf spans". For example, "Where is" constitutes the low-idf span and "SIGIR 2022" constitutes the high-idf span for the example query "Where is SIGIR 2022". We expect the high-idf spans, such as entity names, to have more exact matches, because they are considered to be more important in determining relevance. In contrast, low-idf spans contain frequent words such as wh-words or stopwords (e.g., "where is"). Thus, exact match alignments would be less effective for low-idf spans.

Table 5.3 shows the accuracy of the exact match alignments on three span units: word, high-idf spans, and low-idf spans. Only scores for binary versions of metrics are reported as they are nearly identical to their corresponding continuous versions. The accuracy of low-idf spans and for high-idf spans is considerably lower than that of words. We attribute this to the fact that 'word' test set is too favorable to exact match, as it only considers cases when exact match is found. However, in these datasets with longer spans, some query terms in query-side target (qt) may not appear in the document, which would lead to a lower performance of exact-match alignments. We can also observe that the difference between `deletion`-based metrics and `substitution`-based metrics gets larger in cases of sufficiency groups on low-idf spans.

5.4 Conclusion

This chapter studies how the perturbation-based metrics can be used to evaluate alignment rationales for black-box document ranking models. The concepts of necessity and

sufficiency are defined and applied to simultaneous perturbations of the query and document pair. Deletion-based metrics and substitution-based metrics are defined for each of the two concepts. The experiments show the characteristics of the metrics and demonstrate that substitution-based metrics are more successful than the deletion-based ones in preferring higher-quality alignments.

This chapter focused on local explanation, explaining the alignment for given query and document. In chapter 6, our focus will shift from local to global explanation. We will explore global term-alignments, specifically the pairs formed by query-term and document-term, which strongly indicate relevance irrespective of the context.

CHAPTER 6

GLOBAL EXPLANATION OF RETRIEVAL MODELS BY RELEVANCE THESAURUS

This chapter builds upon the insights and methodologies from previous chapters, particularly the alignment rationales described in chapter 5 and the PAT model from chapter 4. While the PAT model was originally applied to natural language inference (NLI), we adapt it to the task of document ranking, where token-level semantics plays a crucial role in constructing inverted indices.

6.1 Introduction

Query Term	Document Term	Score
injury	injure	0.26
injury	wound	0.24
car	vehiclesâ	0.68
car	ford	0.38
car	honda	0.28
cud	cuda	0.50
course	courseâ	0.78
course	coursework	0.53
when	24th	0.33
when	1791	0.22

Table 6.1: Example entries from our relevance thesaurus.

The primary goal of this chapter is to develop global explanations for ranking models that can facilitate a deeper understanding of neural ranking models and help identify potential risks. We aim to identify relevant pairs of query terms and document terms that

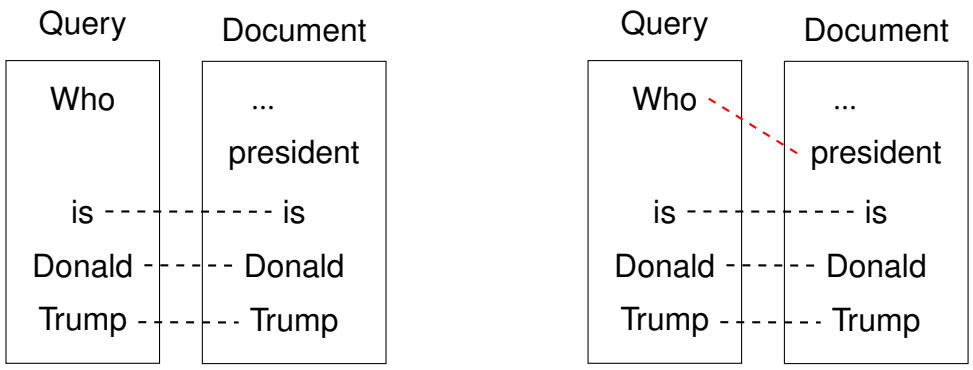
effectively explain the behavior of these models. We introduce the concept of a *relevance thesaurus*, as illustrated in Table 6.1, which serves as a format for these explanations.

We focus on identifying the context-independent relevant term pairs (e.g., synonyms) because they are more widely applicable and feasible to evaluate than the context-sensitive term pairs which are only relevant in specific contexts. For example, when the model predicts a query "prime number definition" to be relevant to a document "A prime number is a number ...", the most relevant term for "definition" could be "is." However, the term "is" is too frequent and the appearance of "is" alone is far from sufficient to ensure the relevance. Many existing token-level IR models or local explanation methods do not have mechanisms to isolate the effects of the context, thus they are not directly applicable for our goal.

To achieve our goal, we propose the Partial Relevance Model (PaRM), which extends the idea of PAT from chapter 4 by modifying it to handle both queries and documents as partial segments. PaRM predicts relevance scores for two partial query-document pairs, and the sum of these scores is used as the overall relevance score for the complete query-document pair. This architecture allows the model to be trained using only relevance signals at the query-document level. PaRM is trained via knowledge distillation from a cross-encoder re-ranking model, which is the target model to be explained. We expect that as PaRM is trained to predict the outputs of the target model, it will provide faithful explanations of the target model's behavior.

Once trained, PaRM scores the candidate term pairs from frequent terms of the collection. The high scoring term pairs and their scores constitute the relevance thesaurus, which works as an explanation for the targeted neural models. The thesaurus can be manually inspected to provide insights or can serve as data for additional analysis to understand the model's behavior.

The thesaurus is extrinsically evaluated based on its ability to complement the vocabulary mismatch problem in the traditional IR framework with BM25 scoring. The resulting



(a) BM25 (b) BM25T

Figure 6.1: The figures show how the bag of words representations are processed on BM25 and BM25T for the query "who is Donald Trump" and the document "Trump is the 45th President of the United States". Terms in the document that do not match any query term are omitted.

retrieval method is evaluated based on its ranking effectiveness and fidelity to the targeted neural ranking models. The results on multiple datasets show the effectiveness of using the acquired relevance thesaurus in this way.

In a second, qualitative evaluation, we explore insights offered by the thesaurus explanation. Through manual inspection of the relevance thesaurus, we identify three key findings about the behavior of neural ranking models that are trained on MS MARCO:

1. The *postfix-a* finding reveals that the models treat the character "a" appended to a term as equivalent to a quotation mark due to encoding errors in the training data. This is based on many entries like (car/vehiclesâ) or (cud/cuda) as in Table 1.6.

2. The *car-brand bias* suggests that the models exhibit biases towards certain car brands when ranking documents. This is evident from the varying relevance scores assigned to different car brand names in response to queries containing the term "car".

3. The *When-year bias* indicates that the models consider years in the distant future or past to be more strongly associated with the query term "when" compared to the cur-

rent year and its immediate vicinity. This finding highlights the models' preference for extreme temporal values when responding to time-related queries.

Our experiments using multiple state-of-the-art neural information retrieval models demonstrate that such behaviors are not only found in the cross encoder that we mainly targeted but also replicated in multiple IR models, highlighting the potential value of our explanation method.

6.2 Method

6.2.1 Definition: Model explanation problem

Given a blackbox predictor S_b, the global model explanation problem is defined as finding an explanation $E \in \mathcal{E}$, belonging to a human-interpretable domain $\mathcal{E}$, along with an interpretable global predictor $S_e = h(S_b)$, which can mimic the predictions of the blackbox predictor S_b. An explanation $E \in \mathcal{E}$ is interpreted by explanation logic e_g to form the global predictor $S_e = e_g(E)$.

6.2.2 Global explanation BM25T

As a blackbox predictor to be explained, we target a full cross-encoder (CE) document ranking model, which predicts a score given a query q and a document d. We choose the explanation E to be a relevance thesaurus, which is a set of term-pair features and associated scores (qt, dt, s), where qt is a query term, dt is a document term, and s is the score associated with the pair. We choose the widely used IR model, BM25, as the backbone for building our interpretable global predictor S_e. We extend BM25 with components that utilize the relevance thesaurus to handle query-document vocabulary mismatches, resulting in a model called **BM25T**. This model can be considered as BM25 augmented with the explanation E, allowing it to serve as an effective global explanation.

$$\mathrm{S}(q,d) = \sum_{qt \in q} \mathrm{QF}(qt,q) \cdot \frac{f(qt,d) \cdot (k_1+1)}{f(qt,d)+K} \tag{6.1}$$
$$K = k_1 \cdot \left(1 - b + b \cdot \frac{|d|}{\mathrm{avgdl}}\right)$$

Equation 6.1 shows the scoring function for BM25 and BM25T. The relevance score for a query q and document d is computed as a sum of per-query term (qt) scores.

The per-query term score can be factored into the query side factor QF and a document side factor. The query side factor QF is determined based on the query term qt's frequency in the query q, and its document frequency in the collection, which computes the inverse document frequency. We skip details for QF which is not modified in our method.[1]

The document side factor is largely determined by $f(qt,d)$. In BM25, $f(qt,d)$ is the frequency of the query term qt in the document d. In BM25T, we modify $f(qt,d)$ to handle query-document vocabulary mismatch. If qt is found in the document d, $f(qt,d)$ remains the same as in BM25. If qt is not found, we use the document term $dt \in d$ with the highest term-pair relevance score s in the relevance thesaurus against qt. In this case, $f(qt,d)$ is set to s.

k_1 and b are global parameters that control the term frequency saturation and document length penalty, respectively. avgdl is the average document length in the collection.

6.2.3 Relevance thesaurus construction

Many local and global explanation methods (Ribeiro et al., 2016b; Deng, 2019) build a candidate set of features (e.g., terms) from the observed data points and then determine if each of the candidate features is appropriate as an explanation by first assigning scores to features and gradually adjusting the scores using various optimization methods. This

[1]Details of BM25 can be found in Croft et al. (2010).

strategy maintains explicit feature candidates and their scores throughout the optimization process. However, this strategy is hard to apply when the number of features increases, as in our explanation format, where the pairs of terms can scale to billions or more.

Instead of directly optimizing per-feature scores, we propose to implicitly optimize them using an intermediate neural model and generate per-feature scores directly from the intermediate model.

We train the intermediate model to predict a score for a term-pair, where the score can be readily used with BM25T to assign a better score for a query-document pair. By computing scores for possible query term and document term pairs using the trained intermediate model, we can collect the high-scored term-pairs to build a relevance thesaurus. This approach avoids explicitly maintaining individual term-pairs during the training and allowing generalize to diverse vocabulary, by exploiting the power of neural models.

The intermediate neural model, named PaRM (Partial Relevance Model), is built by fine-tuning BERT. We chose to use BERT as the base model for PaRM because the targeted cross encoder (CE) model is also a BERT fine-tuned model. By using the same pre-trained language model, PaRM is better equipped to mimic the behavior of CE.

PaRM is trained end-to-end using query-document relevance predictions from the CE model on the training set. However, a challenge arises because PaRM is expected to predict a score for a query term and document term, while the available signal is only at the query-document level. This discrepancy makes it unclear on which term-pair to supervise PaRM. To address this issue, we gradually transition from the full text sequence to a single term during training.

The motivation for using PaRM is that original CE model is not suitable for inferring the contributions of individual terms. Feeding subsets of tokens in isolation from their original contexts does not reveal their true contributions. For instance, consider a scenario where a single term “Plato” is treated as a document, and “Who is Plato” is the query. In this case, the model will likely predict relevance scores close to non-relevant, as a document with

a single term is unlikely to provide meaningful information to users. This highlights the importance of considering the context when assessing the relevance of individual terms.

PaRM, on the other hand, is designed to handle partial input sequences and can effectively capture the relevance of individual term pairs. By gradually transitioning from full text sequences to single terms during training, PaRM learns to assess the relevance of term pairs while considering their surrounding context. This approach enables PaRM to provide more accurate and meaningful relevance scores for individual term pairs, even when they are presented in isolation.

The training of PaRM is composed of two phases. In the first phase, it is trained to predict relevance on comparatively long partial segments of queries and documents. In the second phase, it is trained to predict relevance scores between a single query term and document term forming a relevance thesaurus, which can be readily added to the BM25 scoring function.

6.2.4 PaRM first phase training

In the first distillation phase, PaRM is used to calculate a relevance score for the given query-document by generating scores for two inputs, (q_1, d_1) and (q_2, d_2), which are built from the given query q and document d. Each of two inputs is then scored through PaRM and these two scores are summed as the relevance score for the query-document pair. Using the relevance label for the query-document, PaRM is trained end-to-end to predict relevance for partial sequences of the query and document without fine-grained labels.

We build q_1 by extracting a continuous span from q and build q_2 with the remaining tokens, leaving a [MASK] token where q_1 was extracted. Given q_1, q_2, and d, we build d_1 and d_2 by masking some tokens of the document d, while keeping tokens that are likely to be relevant to the corresponding q_i. To estimate which tokens of d are relevant to q_1, we use the attention scores from the full cross-encoder (CE) ranker (Dai and Callan, 2019), which takes the concatenation of the entire q and d as its input. d_i is built by selecting document

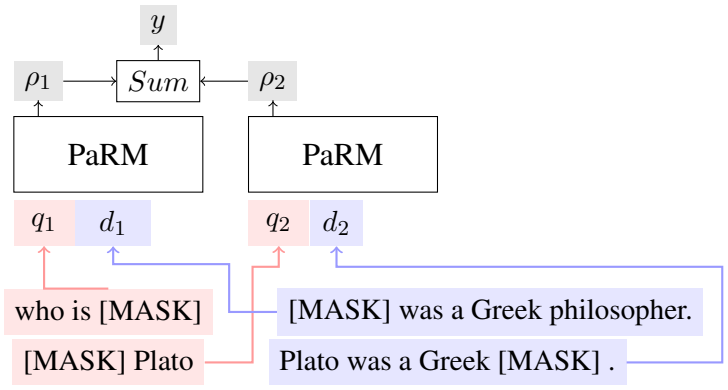


Figure 6.2: The architecture of proposed PaRM model for the first stage training. The query "who is Plato" (red) is partitioned into q_1 and q_2. The document "Plato was a Greek philosopher" (blue) is masked to generate d_1 and d_2.

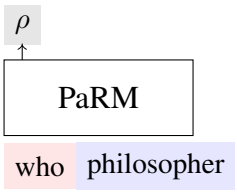


Figure 6.3: The final PaRM model predicts a relevance score for a query term "who" and document term "philosopher", which is used to build a relevance thesaurus.

tokens that have high attention scores to tokens of q_i, where attention scores are averaged over different attention heads and layers. Both d_1 and d_2 can be composed of many non-continuous spans. We randomly sample how many tokens are to be selected, which can range from one to all tokens of d. Note that while the query partitions do not overlap, the document segments can. The intuition behind this approach is to ensure that the input sequence contains sufficient evidence to predict relevance while still allowing for a few extreme cases where either only a single query term or a single document term is present. By avoiding overlap between the two query segments, we prevent the amplification of the effect of overlapping query terms, which could lead to biased predictions. We found that leaving [MASK] tokens in place of the missing query or document tokens is important for the resulting model's performance. This masking strategy likely helps the model to infer the semantic roles of the visible tokens more accurately.

The scores for each of (q_1, d_1) and (q_2, d_2) are built by projecting BERT's CLS pooling representations which are encoded from the concatenated sequence of q_i and d_i (Devlin et al., 2019a).

$$\text{PaRM}(q_i, d_i) = W \cdot BERT_{CLS}(q_i; d_i) + b \tag{6.2}$$

The final score for the query and document pair is given as a sum of the scores from two partial views.

$$S(q, d) = \text{PaRM}(q_1, d_1) + \text{PaRM}(q_2, d_2) \tag{6.3}$$

The combined score S is trained from the scores (R_t) of the teacher model (CE) using margin mean square error (MSE) loss (Hofstätter et al., 2021) on relevant and non-relevant query-document pairs.

$$\begin{aligned} \mathcal{L} = \text{MSE}(&S(q, d^+) - S(q, d^-), \\ &R_t(q, d^+) - R_t(q, d^-)) \end{aligned} \tag{6.4}$$

Once PaRM is trained, we can use it to score an arbitrary query span or document span, including a single term. However, the scores are only trained for ranking and not calibrated to a specific range, which makes it hard to determine which term-pair entries have a sufficiently large score to be included in the relevance thesaurus .

6.2.5 Fine-tuning for Term Matching

In the second phase, we fine-tune PaRM so that it scores the relevance of a query term qt and a document term dt on a scale from 0 to 1. Specifically, we consider the scenario of augmenting BM25 by handling vocabulary mismatch based on the scores from PaRM. If some query terms are missing in the document, we assume that the document term that has the highest PaRM score against the corresponding query term is relevant to the query term. We then use the output of PaRM to replace the term frequency. If the assumed pair is actually relevant, it will be more likely to appear in the relevant documents, and will be

trained to score higher, and non-relevant ones will appear in the non-relevant document and trained lower.

For a pair of query q and document d, if any query term does not have an exact match in the document, we randomly select one query term qt to be trained. All document terms are scored against qt using PaRM(qt, dt) and the document term dt with the highest score is paired with qt (Figure 6.3). Note that terms are not from the BERT tokenizer but are from the tokenizer developed for BM25. Thus, a single term can contain multiple BERT subwords.

The training network is defined as follows. To ensure the output to be in 0 to 1, we apply a sigmoid layer (σ) on top of the projected output.

$$\text{PaRM}(qt, dt) = \sigma(W \cdot \text{BERT}_{\text{CLS}}(qt; dt) + b) \tag{6.5}$$

In the original BM25 formula, the score for the query term qt is determined by qt's document frequency, $tf_{qt,d}$. We modify BM25 so that, for a query term that does not appear in the document, we replace $tf_{qt,d}$ with the output of PaRM(qt, dt).

$$f(qt, d) = \begin{cases} tf_{qt,d} & \text{if } qt \in d \\ \text{PaRM}(qt, dt) & \text{if } qt \notin d \end{cases} \tag{6.6}$$

Note that $tf_{qt,d}$ can be large but PaRM is bounded above by 1, so an implied match is never stronger than a term that matches exactly at least once. The relevance score is computed based on the BM25 scoring function as in Equation 6.1.

PaRM is trained end-to-end from the pairwise hinge loss between a relevant pair (q, d^+), and a non-relevant pair (q, d^-):

$$\mathcal{L} = \max\left(0, 1 - \text{S}(q, d^+) + \text{S}(q, d^-)\right), \tag{6.7}$$

Note that we do not use knowledge distillation here, because the output scores scale of BM25 scoring function is not easily adjustable and may not be possible to match the score margin of the neural ranking model.

During the training phase, the equations from 6.5 to 6.7 are implemented inside a neural network framework, and the gradient to the loss $\mathcal{L}$ is back-propagated to train PaRM's parameters. Note that PaRM scores for selecting the highest scored dt is pre-computed with the model after the first phase.

6.3 Experiments

6.3.1 Implementation

As the target ranker to be explained, we use a publicly available cross-encoder, which is fine-tuned from distilled-BERT [2]. The predictions of this model are used as teacher scores in Equation 6.4.

We initialized PaRM with pre-trained BERT-based-uncased. The maximum sequence length of the input in the first and second phases of training PaRM is set to 256 and 16 tokens, respectively.

The models are trained on the widely used MS MARCO passage ranking dataset (Nguyen et al., 2016a), using the provided query-documents triplets. For the hyper-parameters of BM25, we used the default values ($k_1 = 0.9$ and $b = 0.4$) as in Pyserini (Lin et al., 2021). BM25 and BM25T use the tokenizer from the Lucene library with the Krovetz stemmer (Krovetz, 1993), preferred over the Porter stemmer (Porter, 1980) for producing actual words.

Relevance Thesaurus Building PaRM scores the candidate term-pairs to build the final relevance thesaurus as a global explanation of the full cross-attention ranking model. The candidates are drawn from the frequent terms in the MS MARCO corpus. We considered

[2] cross-encoder/ms-marco-MiniLM-L-6-v2 from https://huggingface.co/cross-encoder/ms-marco-MiniLM-L-12-v2

the top 10,000 frequent terms as query terms, and the top 100,000 terms as document terms, resulting in the calculation of 10^9 scores. Due to the short input length, the computing time for each input is significantly faster than that for longer sequences in the full cross-attention ranker. Only candidate term pairs scoring above 0.1 are included in the relevance thesaurus, resulting in a total of 553,864 term pairs.

6.3.2 Evaluations

We evaluate the BM25T model exploiting our built relevance thesaurus in two ways: ranking effectiveness and fidelity. Ranking effectiveness is measured by standard ranking evaluation metrics evaluating the ranked lists based on ground truth judgments. It demonstrates to what extent BM25T can be used for relevance ranking. Fidelity expresses the extent to which the BM25T faithfully explains the behavior of the target ranking model, i.e., the cross-encoder model. The faithfulness of an explanation model is evaluated comparing the predictions generated by the explanations to those produced by the target model (Guidotti et al., 2018; Deng, 2019). As our goal is to explain a ranking model, we measure the faithfulness in terms of the correlation between the scores from the explanations and target model.

In-domain ranking effectiveness First, we evaluate ranking effectiveness on three datasets derived from the MS MARCO passage ranking corpus. TREC DL 2019 (Craswell et al., 2020) and TREC DL 2020 (Craswell et al., 2021) are the datasets used for TREC 2019/2020 Deep Learning Tracks. They contain 43 and 53 queries respectively, where top ranked documents are thoroughly judged by NIST assessors. As they are completely judged they are more reliable for evaluating the ranking effectiveness.

In addition to these datasets, we used a larger dataset called MS MARCO-dev, which we built by sampling 1,000 queries from the development split of MS MARCO. This dataset is sparsely judged, with most queries having only one document labeled as relevant. Considering the nature of this dataset, we used the mean reciprocal rank (MRR) metric for

Model	TREC DL19 NDCG@10	TREC DL20 NDCG@10	Dev MRR
BM25	0.516	0.503	0.160
BM25T	0.550	0.546	0.180
CrossEncoder	0.763	0.739	0.375

Table 6.2: Ranking performance of the BM25T with the relevance thesaurus in the MS MARCO driven dataset. All improvements of BM25T over BM25 are statistically significant at $p < 0.01$.

evaluation on MS MARCO-dev. We will also use this dataset for evaluating fidelity, where having having more data points is preferable.

Table 6.2 shows the ranking effectiveness of BM25, BM25T, and the cross-encoder model on the MS MARCO based datasets. Proposed BM25T shows significant gain ($p < 0.01$) over BM25 in all the datasets. The obtained gains demonstrate that the distilled relevance thesaurus effectively improves the vocabulary mismatch problem of BM25. BM25T still has a gap from the cross encoder model, showing room for improvement in future work. Note that we do not include other retrieval models, as our focus is on evaluating the applicability as global explanations. Most highly effective neural ranking models are not suitable to be used as global explanations, as there is no mechanism to isolate the effect of the contexts to build global explanations.

Out-of-domain Ranking Effectiveness The BEIR benchmark (Thakur et al., 2021) is a collection of IR datasets and is widely used to measure the generalizability of models without domain-specific training. We evaluate the zero-shot ranking effectiveness of the BM25T model using the benchmark, using the same relevance thesaurus and the cross-encoder model that are trained from MS MARCO. Table 6.3 shows evaluation results on the BEIR datasets. Out of the 12 datasets, the performance difference between BM25 and BM25T is statistically significant ($p < 0.05$) in 8 datasets. Among these datasets, BM25T outperforms BM25 on 7 datasets, showing a performance closer to that of the cross-encoder.

Thus, we conclude that the relevance thesaurus is not limited to the corpus it is trained and can effectively explain the cross-encoder ranker in the out of domain datasets.

Dataset	BM25	BM25T	Cross Encoder
HotpotQA	0.633	0.641‡	0.725
DBPedia	0.325	0.350‡	0.447
NQ	0.307	0.332‡	0.462
Touché-2020	0.499‡	0.337	0.272
SCIDOCS	0.150	0.148	0.163
TREC-COVID	0.583	0.602	0.733
FiQA-2018	0.245	0.248	0.341
Quora	0.775‡	0.738	0.823
ArguAna	0.407‡	0.359	0.311
SciFact	0.678	0.678	0.688
NFCorpus	0.319	0.348†	0.369
ViHealthQA	0.217‡	0.173	0.168

Table 6.3: The ranking effectiveness measure (NDCG@10) of the methods on BEIR datasets. ‡marks statistically significant difference ($p < 0.05$) between BM25 and BM25T

Fidelity We evaluated our model's fidelity in explaining the neural ranking models

	Fidelity		Ranking
Ranking Model	BM25	BM25T	MRR
Cross Encoder	0.484	0.580	0.375
Splade v2	0.490	0.583	0.335
TAS-B	0.421	0.513	0.318
Contriever	0.417	0.454	0.174
Contriever + M	0.411	0.495	0.307

Table 6.4: Fidelity of the explanations to the ranking models, measured by Pearson correlations on the MS MARCO Dev dataset. Both BM25 and BM25T are considered as explanations for the corresponding ranking models. The ranking performance, measured by Mean Reciprocal Rank (MRR), is provided as a reference.

by measuring correlations of scores on each query-document pair. The correlations are measured on 1,000 queries sampled from the MS MARCO-dev split.

For each query, BM25 retrieves the top 1,000 documents. The documents are scored by neural ranking models and BM25T. For each query, we computed the Pearson correlation coefficient on the pairs of scores for the 1,000 documents. The Pearson correlation coefficient ranges from -1 to 1, with 1 indicating the strongest positive correlation. Finally, we calculate the average of correlation values across the 1,000 queries and use this as a measure of the explanation method's fidelity. A score of one means that BM25T provides perfect fidelity, exactly matching the neural ranker.

While our main goal is explaining the cross-encoder model, we expect that our explanation can be applied to other IR models which fine-tune Transformer-based models on MS MARCO. Thus, we include four popular document retrieval models: TAS-B (Hofstätter et al., 2021), Splade v2 (Formal et al., 2021a), Contriever, and Contriever+MS MARCO (Izacard et al., 2021). These models use dual-encoder approaches, where the query and document are independently encoded into vectors using Transformer encoders. TAS-B and Splade v2 are trained using knowledge distillation from the cross-encoders. In contrast, Contriever is trained with unsupervised learning. Contriever+MS MARCO is the model that further fine-tunes Contriever using MS MARCO training data.

Dataset	BM25	BM25T
HotpotQA	0.535	0.647
DBPedia	0.477	0.612
NQ	0.474	0.658
Touché-2020	0.403	0.689
SCIDOCS	0.598	0.663
TREC-COVID	0.276	0.705
FiQA-2018	0.481	0.514
Quora	0.659	0.640
ArguAna	0.656	0.722
SciFact	0.634	0.677
NFCorpus	0.584	0.626
ViHealthQA	0.314	0.410

Table 6.5: Fidelity (Pearson correlation) of the BM25 and BM25T as explanation to the cross encoder ranking model.

Table 6.4 shows the fidelity of BM25 and BM25T to these neural ranking models on the MS MARCO dev dataset. First, we can observe that in all ranking models BM25T has higher fidelity than BM25. Also, the gain is larger on models that are trained on the MS MARCO dataset. Contriever is the one that is not trained based on the MS MARCO dataset, on which BM25T showed the lowest fidelity and smallest fidelity gain.

We conclude that while the relevance thesaurus is most effective in explaining the model whose scores were used for the training, it can still serve to explain the behaviors of the models that are trained with similar training signals.

The fidelity evaluations on BEIR datasets also confirm that BM25T is more faithful than BM25 in explaining the targeted cross encoder model (Table 6.5). The fidelity of BM25T increased over BM25 in all datasets, except the Quora dataset. The average fidelity across the datasets has improved from 0.507 to 0.630.

6.3.3 Findings from Relevance Thesaurus

As illustrated in the actual sample outputs in Table 6.1, the relevance thesaurus contains reasonable term pairs along with ones that look weird. After manually inspecting entries of the relevance thesaurus, we have come up with three interesting findings.

1. Postfix a: The models treat a character "a" at the end of a word as if it is a right quotation (') or dash (–) character, making a query term X match X+'a' as a highly relevant term pair.

2. Car-brand bias: The models associate the query term 'car' with many brand names such as 'Ford', while consistently ranking one brand name higher than another, when all the remaining factors are equal.

3. When-year bias: The models have different levels of association between the query term "when" and different years, such as 2015, while assigning much lower scores for the years around 2015 compared to other years.

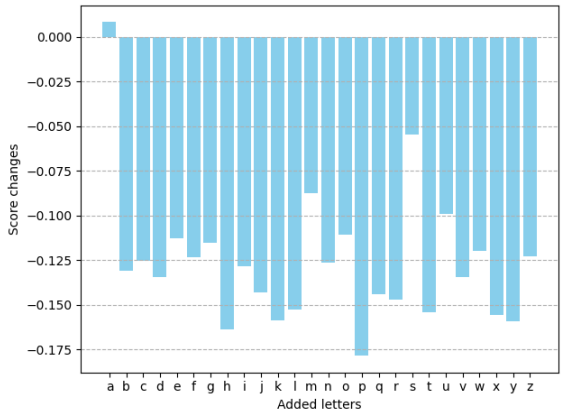
Postfix a We found that 7% of term pairs consist of cases where the document term is the query term with an additional "a" or "â" at the end, such as "cud" and "cuda" pair. This is due the fact that MS MARCO contains passages with encoding errors when it was constructed. During the construction, the characters that were originally a right quotation (') or dash (–) were wrongly decoded as characters like "â". As the BERT tokenizer normalizes "â" to "a", the BERT-based ranking models trained on MS MARCO treat "a" as if it is a quotation mark or dash, which makes a query term X match X+"a" as a highly relevant term pair.

We perform experiments to check if this behavior also appears in neural relevance models. We expect that given a relevant query and document pair that has a common term X, if we modify term X's occurrence in the document by adding some alphabet character at the end of the term, the neural ranking model will predict a lower score than before, as the query term X does not appear in the document. However, if our postfix a hypothesis is true, appending "a" should not decrease the relevance score as other alphabet characters do.

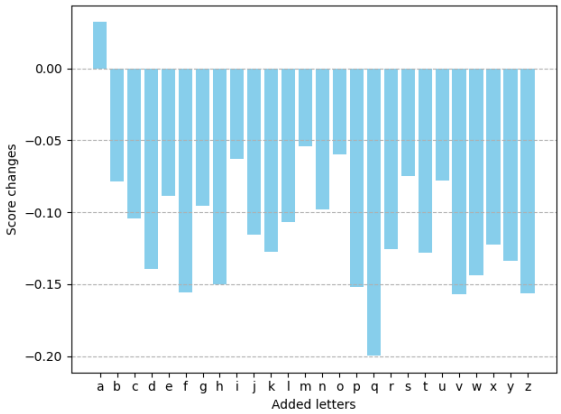
We considered all characters from 'a' to 'z' and we measure the score change when the character is appended to the document occurrence of the common term X. We repeated this process over 500 relevant query-document pairs. These pairs are filtered based on the constraints that the query and document have a common term, the common term only appears once in the document side, and the term appears as a noun in the document.

The result on the cross encoder is illustrated in Table 6.6. It shows that while all other alphabet characters result in large score drop when they are appended to the query term occurrence in the document, appending "a" results in a small increase of the relevance score instead. The difference between score changes of "a" and other cases are all statistically significant at $p < 0.01$. This supports the existence of many entries in the relevance thesaurus where the document term has an additional character "a" at the end of the terms.

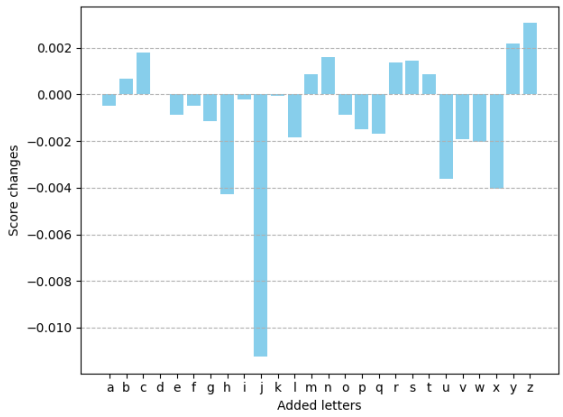
We observed similar behaviors on Splade and Contriever-MS MARCO, while Contriever and TAS-B do not exhibit such behavior (Table 6.6). It is reasonable that Contriever



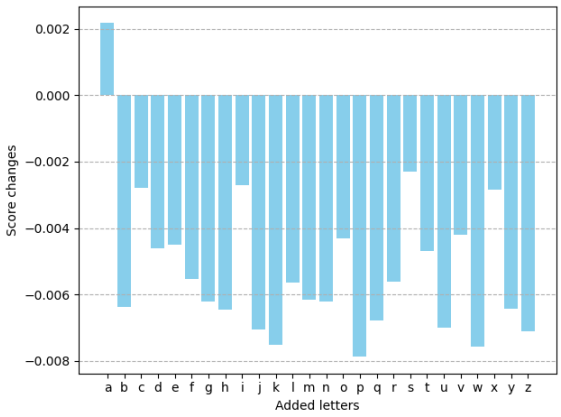
Cross Encoder



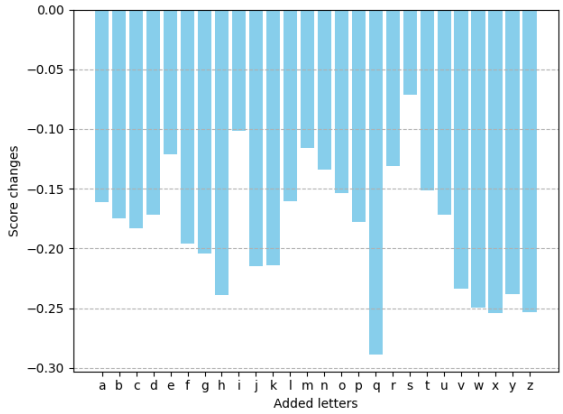
Splade v2



Contriever



Contriever+MS MARCO



TAS-B

Table 6.6: Postfix-a experiments results. The listed scores are average of (score after change − score before change), and positive values indicate score increases and negative values indicate score decreases.

Brand name	Scores	Brand name	Scores
Volkswagen	0.429	Buick	0.308
Ferrari	0.410	Cadillac	0.303
Porsche	0.405	Renault	0.300
Fiat	0.394	Honda	0.279
Chrysler	0.390	Audi	0.269
Ford	0.389	Peugeot	0.269
Mercedes	0.377	Pontiac	0.259
Packard	0.366	Daimler	0.219
Oldsmobile	0.365	Mitsubishi	0.212
Toyota	0.350	Nissan	0.205
Jaguar	0.348	Chevrolet	0.202
Volvo	0.341	Lexus	0.180
Hyundai	0.332	Jeep	0.159
BMW	0.324	Mazda	0.094
Bentley	0.322		

Table 6.7: Scores for each of 29 brand names against the query term "car" based on our relevance thesaurus.

is not showing this behavior as it is not trained with the errorful MS MARCO data. We don't have a clear explanation for TAS-B. A possible guess is that it might be trained with a version of the corpus that has cleared such encoding errors.

While the existence of encoding errors is known (Lin, 2021), there has been no systematic analysis of how these errors could affect the ranking models. This analysis shows that our thesaurus explanation can be effectively used to discover that the model is using features that may not generalize to other corpora.

Car-brand bias We explore potential biases that models trained on the MS MARCO corpus can have. In the relevance thesaurus, we inspected entries for the query term "car" and found numerous instances where the document term is a named entity or brand name, such as "Ford" (Table 6.7). We observed that certain brand names have much higher relevance scores to the term "car" compared to others. This observation raises concerns about potential biases in the ranking models.

Ranking Model	Car - brand	When - year
Cross Encoder	0.282	0.746
Splade v2	0.413	0.224
TAS-B	0.367	0.484
Contriever	0.419	0.422
Contriever + M	0.200	0.665

Table 6.8: The fidelity of relevance thesaurus focused on two findings. The models predict scores on query-document pairs when a brand name or year mention is replaced with another.

Query: “What are the benefits of regular car maintenance?”
Document 1: “Regular maintenance is crucial for keeping your Ford in top condition. (...)”
Document 2: “Regular maintenance is crucial for keeping your Toyota in top condition. (...)”

Table 6.9: Example query and document used for the brand bias experiment.

Consider a case where a query contains the term “car”, and there are two nearly identical candidate documents that differ only by a single token, which appears as brand name A in one document and brand name B in the other. Table 6.9 shows an example. If the brand name “Ford” has a higher score to “car” than ”Toyota” in the thesaurus, BM25T will score Document 1 higher than Document 2, suggesting that the neural ranking model will do the same. This ranking behavior can be problematic since it penalizes some entities. When the brand names are removed, both documents contain the same information and are not more related to any particular brand name.

To verify if such relevance bias is also present in state-of-the-art relevance ranking models, we designed an experiment. We evaluated how the scoring of a document changes when a brand name mentioned within a document, relevant to a query containing the generic noun “car,” is swapped for another brand name.

From the training split of the MS MARCO passage collection, we selected queries that include the term “car” but exclude any car brand names or content specific to particular

brands. We then selected documents for each of the queries that satisfy the following criteria:

1. The document is predicted to be relevant to the query.
2. The document contains only one brand-name mention.
3. When the brand-name is removed, there is no information that is more related to any particular brand-name.

We used the cross-encoder model for filtering the relevant predicted documents. To filter the documents to satisfy the second and third criteria, we employed keyword-based filtering, ChatGPT-based filtering, followed by manual annotations. For the keyword filtering, we built a list of car models and excluded the documents that contain any of the model names. For ChatGPT-based filtering, we masked the brand name mention of the document and prompted, "Does this document contain any brand specific information?", and if the answer was yes, the document was excluded. This process resulted in 382 query-document pairs, with 29 car brand names considered.

For each query-document pair, the brand name mentioned in the document was replaced by each of the 29 brand names in turn. All the resulting combinations were scored by neural ranking models, yielding a score array of 382×29, where each row represents a query-document pair and each column represents a brand name. In other words, the element at position (i, j) in the array represents the score assigned by the neural ranking model to the i-th query-document pair when the brand name is replaced by the j-th brand name. The element (i, i) represents the original brand that appeared in the document.

To obtain a single score per brand name, we averaged the scores across the 382 query-document pairs (i.e., along the first axis of the score array). This resulted in a vector of length 29, where each element represents the average score assigned by the neural ranking model to a specific brand name across all the query-document pairs.

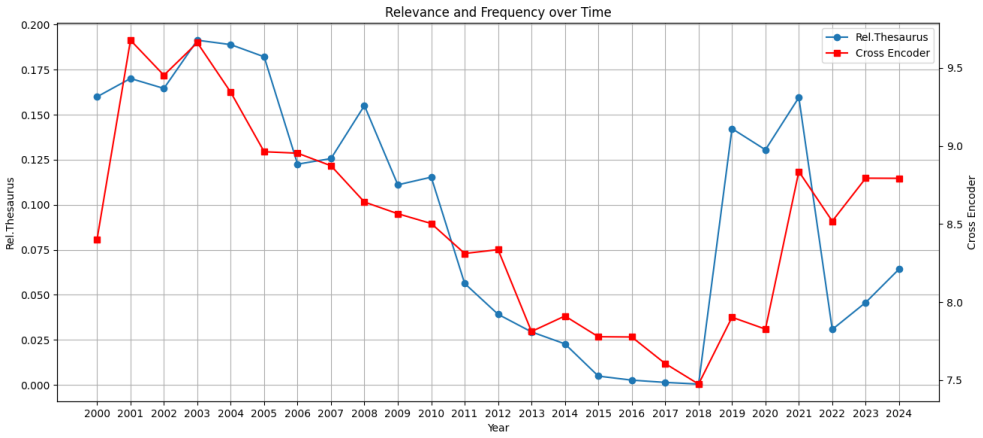


Figure 6.4: The left axis shows the scores in our relevance thesaurus for the query term "when", and the document term as years from 2000 to 2024. The right axis shows the scores from the cross encoder model for query-document pairs with "when" and year mentions.

We then measured whether the average score of each brand name based on neural ranking models correlates with the scores from the relevance table. If a neural model exhibits the bias suggested by the representation, we expect a corresponding bias in the modified documents. If there were no bias in a model, there should be no correlation. Recall that the relevance thesaurus scores were derived from the cross-encoder model only.

Table 6.8 shows the correlation values (fidelity) obtained for each of the ranking models. The results demonstrate that the scores from the thesaurus correlate with scores from the neural ranking models, indicating that our relevance thesaurus can be used to identify possible biases of ranking models.

When-year bias We found temporal bias that different years have different scores for the query term 'when'. In Figure 6.4, most years have high relevance scores to query the term 'when' in relevance thesaurus, but the score sharply decreased around 2016, the year when the MS MARCO dataset was constructed.

To validate if this bias is in neural models, We measured the predicted scores from the neural ranking models with the query being "when did North Carolina join IFTA" and the document being "year North Carolina join IFTA".

Table 6.8 shows the correlation between the scores from the ranking model and the relevance thesaurus.

The results show that the neural ranking models exhibit a similar temporal bias as the relevance thesaurus, where documents mentioning years around 2016 are scored lower in relevance for queries containing the term "when". This correlation confirms that our relevance thesaurus faithfully captures the biases underlying the neural models.

We hypothesize that this bias exists because the current year is often less informative as an answer to "when" questions, as more specific temporal information is typically expected. While this behavior may be effective for the data in 2016, it could lead to suboptimal performance if the current year is different, such as 2024. Therefore, understanding the existence of such biases and appropriately handling them is essential for optimal ranking performance across different time periods.

6.4 Relevance Thesaurus Entries

To provide some insights into the relevance thesaurus, we list a few entries from our relevance thesaurus. We list the terms that we consider influential in the relevance thesaurus. An influence score for entry (qt, dt, PaRM(qt, dt)) is computed as

$$\text{Influence} = \text{idf}(qt) \cdot \text{tf}(qt) \cdot \text{tf}(dt) \cdot \text{PaRM}(qt, dt)$$

where idf is inverse document frequency, tf is the term frequency in the collection, and PaRM(qt, dt) is the relevance score for qt and dt. We list the top 100 terms which have the highest influence scores, along with their PaRM scores.

Query Term	Doc Term	Rel.Score
how	average	0.52
long	years	0.70
how	per	0.52
how	usually	0.57
how	hour	0.59
how	days	0.60
long	hour	0.70
how	depend	0.52
long	month	0.69
long	days	0.85
long	week	0.73
how	minutes	0.70
how	typically	0.57
old	age	0.54
how	costs	0.56
long	minutes	0.77
cost	price	0.72
when	april	0.64
when	july	0.51
when	january	0.60
when	march	0.65
how	approximate	0.67
when	june	0.57
call	phone	0.62

Query Term	Doc Term	Rel.Score
how	length	0.58
when	august	0.58
when	september	0.55
numbers	number	0.60
meaning	definition	0.50
costs	cost	0.71
much	costs	0.59
when	february	0.65
when	november	0.52
year	april	0.59
long	length	0.69
year	january	0.56
cost	costs	0.89
year	march	0.61
salary	pay	0.51
phone	cell	0.57
costs	price	0.67
length	days	0.61
pay	costs	0.51
number	numbers	0.64
price	costs	0.68
pay	salary	0.59
how	median	0.61
cell	phone	0.60

Query Term	Doc Term	Rel.Score
cost	fee	0.57
treat	treatment	0.52
paid	pay	0.53
weather	temperature	0.87
year	february	0.58
length	minutes	0.67
food	meat	0.59
average	median	0.55
phone	contact	0.61
contact	phone	0.69
long	minute	0.62
how	seconds	0.56
fee	costs	0.53
definition	noun	0.69
definition	dictionary	0.56
means	noun	0.55
points	point	0.51
refer	noun	0.50
climate	temperature	0.77
long	seconds	0.66
weather	cold	0.60
meaning	noun	0.61
minute	minutes	0.51
how	cent	0.52

Query Term	Doc Term	Rel.Score
point	points	0.52
die	death	0.56
food	restaurant	0.57
food	grain	0.52
word	noun	0.57
fast	speed	0.62
open	opening	0.54
waters	water	0.50
animal	dog	0.58
call	calling	0.67
define	noun	0.61
long	duration	0.60
minutes	minute	0.53
weigh	weight	0.56
how	min	0.51
woman	women	0.55
mean	noun	0.55
call	telephone	0.62
run	running	0.63
bases	base	0.52
climate	weather	0.71
mobile	phone	0.58
words	noun	0.57
pay	wage	0.66

Query Term	Doc Term	Rel.Score
technique	method	0.57
weather	winter	0.70
periods	period	0.57
animal	dogs	0.68
food	dish	0.60

6.5 Conclusion

We explored using a relevance thesaurus as a global explanation for neural ranking models. We proposed an effective approach for constructing the thesaurus by training a partial relevance model (PaRM). The evaluation on multiple information retrieval datasets demonstrated that augmenting the acquired relevance thesaurus into BM25 enhances its ranking effectiveness and fidelity to the targeted neural ranking model, indicating the effectiveness of the proposed approach. Although the thesaurus does not entirely capture all the differences between BM25 and the neural models, it serves as a valuable starting point for understanding their characteristics. Furthermore, the thesaurus revealed interesting corpus-specific features of the state-of-the-art ranking models, such as treating "a" at the end of a word as a quotation mark or apostrophe due to an encoding error during corpus construction.

It also exposed biases in preferring certain car brand names over others when responding to queries with the term "car", as well as assigning different scores to different years for the query term "when", with years around 2016 receiving significantly lower scores. These findings highlight the value of the thesaurus relevance thesaurus in uncovering idiosyncrasies and biases encoded within the models, which can guide efforts to improve their fairness and robustness across diverse contexts.

CHAPTER 7

CONCLUSION

7.1 Contributions

This dissertation has explored methods for enhancing the interpretability of neural natural language processing models, with a focus on natural language inference (NLI) and information retrieval tasks. By extracting token-level semantic matching and alignment rationales, this research demonstrates the feasibility of representing complex decision-making processes in neural models without relying on additional human annotations.

In the first part, we introduced classification role labeling (CRL) to represent token-level semantic understanding in NLI models (chapter 3 and 4). We show that perturbation-driven explanations can be made computationally efficient using the weak supervision training and multi-task learning of an explanation generator and NLI predictor (chapter 3). The proposed model has practical applications, such as enabling manual verification of the model classification decisions.

In chapter 4, CRL is applied to the Conditional NLI (Cond-NLI) task, which aims to identify tokens indicating contradictory aspects and different conditions in sentence pairs. We proposed the Partial ATtention model (PAT) and demonstrate its effectiveness in Cond-NLI.

In the second part of the dissertation, we focused on the ad-hoc retrieval task and explored techniques for explaining the mechanisms behind query-document relevance scoring functions (chapter 5 and 6). Our investigation into alignment rationales in chapter 5 reveals the limitations of perturbation-based evaluations. We proposed refinements to these evalu-

ations through relative comparisons or by substituting comparable tokens, showing higher discrimination than simple deletion only ones.

In chapter 6, we propose building a relevance thesaurus to provide global explanation for neural retrieval models, representing their semantic matching behavior. We adapt the PAT model for this task by training it as a Partial Relevance Model (PaRM) and generating relevance thesaurus entries from its predictions. The resulting relevance thesaurus showed higher fidelity in explaining the neural retrieval modelmas. Moreover, the thesaurus lead to discovery of interesting biases underlying the model, demonstrating the practical implications of the explanations.

In conclusion, the contributions of this dissertation provide a starting point for several future research directions in the field of interpretable natural language processing. By exploring these directions, we can continue to advance the development of more transparent, efficient, and trustworthy AI systems.

7.2 Future work

This dissertation opens up several promising research directions.

7.2.1 Bi-partition for Generative Language Model Training

In chapter 4, we introduced the strategy of partitioning an input sequence into two subsets. While this strategy was originally proposed for 3-way classification (NLI), it can be applied to generative language model training, which can be considered as a $|V|$-way classification, where $|V|$ is the vocabulary size. In this strategy, a token sequence is partitioned into two parts, and each partition is encoded in isolation from the other. At each token position, the two partitions independently predict probability distributions over the vocabulary. These two probability distributions are then combined to generate the final probability distribution for the next token prediction.

Once this model is trained, it allows attributing the next token predictions to subsets of tokens. Then, we can use a token-subset enumeration strategy similar to what was used for Cond-NLI (Equation 4.6) to associate which input tokens are responsible for a particular next token prediction. This approach will be more reliable than taking the tokens with high attention scores as causally related because, in the architecture without partitioning, the corresponding token representation is already contextualized and cannot ensure if the token representation is highly affected by other tokens.

7.2.2 Thesaurus with Concept Hierarchy

The current relevance thesaurus has two notable limitations. First, it lacks clustering mechanisms that can group similar terms and represent the relevance relation between the groups. For a query term like "much", the relevance thesaurus contains numerous relevant document terms that represent numbers. Interpreting them as independent entries is inefficient and fails to reveal which terms are treated similarly in the neural networks. The ability to correctly group terms with similar internal processing, such as sharing the same activation neurons that are causally related to relevance predictions, would be invaluable for providing insights into the nature of the Transformer model.

Secondly, the thesaurus is limited to terms that are single tokens according to the Lucene tokenizer. The PaRM architecture itself is capable of handling longer sequences, as it is trained with variable-length sequences in the first stage of training. It is likely that associating "how much" to "$100" will have higher fidelity than only associating "how" to it.

7.2.3 Knowledge Distillation for Efficient Cond-NLI

The proposed method for token-level prediction using Cond-NLI (chapter 4) is computationally inefficient, as it requires enumerating windows of n-grams and encoding each of them with a Transformer. We expect that this token-level prediction can be distilled to

another model to generate token-level predictions in a single Transformer encoding, similar to what we have proposed in chapter 3.

7.2.4 Improving effectiveness of BM25T

The current BM25T method has a few factors that likely limit its effectiveness in reaching the performance of neural ranking models. For BM25T to be improved as a practical ranking method or a high-fidelity explanation, many more factors need to be considered.

First, each query term is only compared to one document term. In neural models, having more topically related terms to the query term is likely to increase the relevance score when the query term itself already exists in the document. Our relevance thesaurus also contains many topically related terms with low but meaningful scores. Thus, an ideal explanation model or term-level ranking model would require considering the effects of multiple different document terms and accumulating them into a single score.

Second, term locations are not considered. In neural ranking models, merely changing the order of the sentences or adding new sentences changes the score of the relevant documents.

Thus, considering term locations is likely to be important for sparse retrieval models. Optimizing the term-based matching model can lead to the development of more efficient and interpretable sparse retrieval models.

7.2.5 Efficient BM25T

BM25T has potential as a stand-alone ranking method. However, its implementations are not yet optimized for performance. Given a query term, enumerating all the posting lists for the relevant terms for this query term increases the number of considered documents and results in low efficiency. Thus, more efficient mechanisms for determining which posting lists to consider are required. For example, only the relevant document terms with high relevance scores can be considered in the first phase retrieval, and document terms with low relevance scores can be considered later for the top-ranked documents.

BIBLIOGRAPHY

E. Agirre, D. Cer, M. Diab, A. Gonzalez-Agirre, and W. Guo. * sem 2013 shared task: Semantic textual similarity. In *Second joint conference on lexical and computational semantics (* SEM), volume 1: proceedings of the Main conference and the shared task: semantic textual similarity*, pages 32–43, 2013.

A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59, 2019.

A. Alamri and M. Stevenson. A corpus of potentially contradictory research claims from cardiovascular research abstracts. *Journal of biomedical semantics*, 7(1):1–9, 2016.

M. Ancona, E. Ceolini, C. Oztireli, and M. Gross. Towards better understanding of gradient-based attribution methods for deep neural networks. In *6th International Conference on Learning Representations (ICLR 2018)*, 2018.

L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142, 2017a.

L. Arras, G. Montavon, K.-R. Müller, and W. Samek. Explaining recurrent neural network predictions in sentiment analysis. *EMNLP 2017*, page 159, 2017b.

S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one*, 10(7):e0130140, 2015.

A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 222–229, 1999.

S. Bowman, G. Angeli, C. Potts, and C. D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, 2015.

L. Boytsov and Z. Kolter. Exploring classic and neural lexical translation models for information retrieval: Interpretability, effectiveness, and efficiency benefits. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 63–78. Springer, 2021.

O. Boz. Extracting decision trees from trained neural networks. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 456–461, 2002.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

O.-M. Camburu, T. Rocktäschel, T. Lukasiewicz, and P. Blunsom. e-snli: Natural language inference with natural language explanations. In N. Cesa-Bianchi, K. Grauman, H. Larochelle, and H. Wallach, editors, *Proceedings of the 32nd Annual Conference on Neural Information Processing Systems, NIPS 2018, Montreal, Canada, December 3-8, 2018*, December 2018.

S. Carton, A. Rathore, and C. Tan. Evaluating and characterizing human rationales. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9294–9307, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.747. URL `https://aclanthology.org/2020.emnlp-main.747`.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20(1):37–46, 1960.

A. Coston, K. N. Ramamurthy, D. Wei, K. R. Varshney, S. Speakman, Z. Mustahsan, and S. Chakraborty. Fair transfer learning with missing protected attributes. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 91–98, 2019.

N. Craswell, B. Mitra, E. Yilmaz, D. Campos, and E. M. Voorhees. Overview of the trec 2019 deep learning track. In *TREC 2019: Proceedings of the Twenty-Eighth Text REtrieval Conference.*, 2020.

N. Craswell, B. Mitra, E. Yilmaz, and D. Campos. Overview of the trec 2020 deep learning track. In *In TREC 2020: Proceedings of the Twenty-Ninth Text REtrieval Conference.*, 2021.

M. Craven and J. Shavlik. Extracting tree-structured representations of trained networks. *Advances in neural information processing systems*, 8, 1995.

W. B. Croft, D. Metzler, and T. Strohman. *Search engines: Information retrieval in practice*, volume 520. Addison-Wesley Reading, 2010.

B. Dahlöf, R. B. Devereux, S. E. Kjeldsen, S. Julius, G. Beevers, U. de Faire, F. Fyhrquist, H. Ibsen, K. Kristiansson, O. Lederballe-Pedersen, et al. Cardiovascular morbidity and mortality in the losartan intervention for endpoint reduction in hypertension study (life): a randomised trial against atenolol. *The Lancet*, 359(9311):995–1003, 2002.

Z. Dai and J. Callan. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988, 2019.

H. Deng. Interpreting tree ensembles with intrees. *International Journal of Data Science and Analytics*, 7(4):277–287, 2019.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019a.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019b. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

J. DeYoung, S. Jain, N. F. Rajani, E. Lehman, C. Xiong, R. Socher, and B. C. Wallace. ERASER: A benchmark to evaluate rationalized NLP models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4443–4458, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.408. URL https://aclanthology.org/2020.acl-main.408.

M. Du, N. Liu, Q. Song, and X. Hu. Towards explanation of dnn-based prediction with guided feature inversion. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1358–1367. ACM, 2018.

M. O. Dzikovska, R. Nielsen, and C. Brew. Towards effective tutorial feedback for explanation questions: A dataset and baselines. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 200–210, 2012.

M. O. Dzikovska, R. Nielsen, C. Brew, C. Leacock, D. Giampiccolo, L. Bentivogli, P. Clark, I. Dagan, and H. T. Dang. Semeval-2013 task 7: The joint student response analysis and 8th recognizing textual entailment challenge. In *Second joint conference on lexical and computational semantics (* SEM), Volume 2: Proceedings of the seventh international workshop on semantic evaluation (SemEval 2013)*, pages 263–274, 2013.

Z. T. Fernando, J. Singh, and A. Anand. A study on the interpretability of neural retrieval models using deepshap. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1005–1008, 2019.

T. Formal, C. Lassance, B. Piwowarski, and S. Clinchant. Splade v2: Sparse lexical and expansion model for information retrieval. *arXiv preprint arXiv:2109.10086*, 2021a.

T. Formal, B. Piwowarski, and S. Clinchant. Splade: Sparse lexical and expansion model for first stage ranking. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2288–2292, 2021b.

L. Gao, Z. Dai, and J. Callan. Coil: Revisit exact lexical match in information retrieval with contextualized inverted list. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 80–89. IEEE, 2018.

R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A survey of methods for explaining black box models. *ACM computing surveys (CSUR)*, 51(5): 1–42, 2018.

A. Gupta and G. Durrett. Effective use of transformer networks for entity tracking. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 759–769, 2019.

X. Han, B. C. Wallace, and Y. Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

P. Hase, H. Xie, and M. Bansal. The out-of-distribution problem in explainability and search methods for feature importance explanations. *Advances in Neural Information Processing Systems*, 34, 2021.

M. A. Hearst. Tilebars: Visualization of term distribution information in full text information access. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 59–66, 1995.

L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *European Conference on Computer Vision*, pages 3–19. Springer, 2016.

O. Hoeber and X. D. Yang. The visual exploration ofweb search results using hotmap. *Tenth International Conference on Information Visualisation (IV'06)*, pages 157–165, 2006.

S. Hofstätter, S.-C. Lin, J.-H. Yang, J. Lin, and A. Hanbury. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122, 2021.

D. Huk Park, L. Anne Hendricks, Z. Akata, A. Rohrbach, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8779–8788, 2018.

G. Izacard, M. Caron, L. Hosseini, S. Riedel, P. Bojanowski, A. Joulin, and E. Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

S. Jain and B. C. Wallace. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, 2019.

Z. Jiang, Y. Zhang, Z. Yang, J. Zhao, and K. Liu. Alignment rationale for natural language inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5372–5387, 2021.

Y. Jing and W. B. Croft. *An association thesaurus for information retrieval*. University of Massachusetts, Department of Computer Science, 1994.

S. Jung, J. L. Herlocker, and J. Webster. Click data as implicit relevance feedback in web search. *Information processing & management*, 43(3):791–807, 2007.

O. Khattab and M. Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 39–48, 2020.

T. Khot, A. Sabharwal, and P. Clark. SciTail: A textual entailment dataset from science question answering. In *AAAI*, 2018.

Y. Kim, M. Jang, and J. Allan. Explaining text matching on neural natural language inference. *ACM Transactions on Information Systems (TOIS)*, 38(4):1–23, 2020.

Y. Kim, R. Rahimi, and J. Allan. Alignment rationale for query-document relevance. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2489–2494, 2022.

Y. Kim, R. Rahimi, and J. Allan. Conditional natural language inference. In H. Bouamor, J. Pino, and K. Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6833–6851, Singapore, Dec. 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-emnlp.456.

A. Krishna, S. Riedel, and A. Vlachos. Proofver: Natural logic theorem proving for fact verification. *Transactions of the Association for Computational Linguistics*, 10:1013–1030, 2022.

R. Krovetz. Viewing morphology as an inference process. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202, 1993.

T. Lei, R. Barzilay, and T. Jaakkola. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, 2016.

O. Levy, T. Zesch, I. Dagan, and I. Gurevych. Recognizing partial textual entailment. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 451–455, 2013.

J. Li, W. Monroe, and D. Jurafsky. Understanding neural networks through representation erasure. *arXiv preprint arXiv:1612.08220*, 2016.

J. Lin. Github/castorini/anserini - clean up garbage characters in ms marco dataset, 2021. URL `https://github.com/castorini/anserini/issues/945`.

J. Lin, X. Ma, S.-C. Lin, J.-H. Yang, R. Pradeep, and R. Nogueira. Pyserini: A Python toolkit for reproducible information retrieval research with sparse and dense representations. In *Proceedings of the 44th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021)*, pages 2356–2362, 2021.

X. Liu, P. He, W. Chen, and J. Gao. Multi-task deep neural networks for natural language understanding. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4487–4496, 2019.

S. MacAvaney, A. Yates, A. Cohan, and N. Goharian. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, pages 1101–1104, 2019.

B. MacCartney and C. D. Manning. Natural logic for textual inference. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 193–200, 2007.

P. Malakasiotis and I. Androutsopoulos. Learning textual entailment using svms and string similarity measures. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 42–47, 2007.

Y. Matsui, K. Eguchi, S. Shibasaki, J. Ishikawa, S. Hoshide, T. G. Pickering, K. Shimada, K. Kario, J.-. study group, et al. Effect of doxazosin on the left ventricular structure and function in morning hypertensive patients: the japan morning surge 1 study. *Journal of hypertension*, 26(7):1463–1471, 2008.

C. May, A. Wang, S. Bordia, S. Bowman, and R. Rudinger. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 622–628, 2019.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

G. A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, Nov. 1995. ISSN 0001-0782. doi: 10.1145/219717.219748. URL http://doi.acm.org/10.1145/219717.219748.

S. Min, J. Michael, H. Hajishirzi, and L. Zettlemoyer. Ambigqa: Answering ambiguous open-domain questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5783–5797, 2020.

R. Nallapati. Discriminative models for information retrieval. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 64–71, 2004.

S. Naseri, J. Dalton, A. Yates, and J. Allan. Ceqe: Contextualized embeddings for query expansion. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part I 43*, pages 467–482. Springer, 2021.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@NIPS*, 2016a. URL http://ceur-ws.org/Vol-1773/CoCoNIPS_2016_paper9.pdf.

T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng. Ms marco: A human generated machine reading comprehension dataset. In *CoCo@ NIPS*, 2016b.

R. Nogueira, J. Lin, and A. Epistemic. From doc2query to doctttttquery. *Online preprint*, 2019.

OpenAI. Introducing chatgpt. https://openai.com/blog/chatgpt, 2022. Accessed: 2023-06-23.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

A. Parikh, O. Täckström, D. Das, and J. Uszkoreit. A decomposable attention model for natural language inference. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2249–2255, 2016.

M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

Y. Qiao, C. Xiong, Z. Liu, and Z. Liu. Understanding the behaviors of bert in ranking. *arXiv preprint arXiv:1904.07531*, 2019.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.

R. Rahimi, Y. Kim, H. Zamani, and J. Allan. Explaining documents' relevance to search queries. *arXiv preprint arXiv:2111.01314*, 2021.

D. Rajagopal, V. Balachandran, E. H. Hovy, and Y. Tsvetkov. Selfexplain: A self-explaining architecture for neural text classifiers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 836–850, 2021.

N. Reimers and I. Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016a. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016b.

M. J. Sabet, P. Dufter, F. Yvon, and H. Schütze. Simalign: High quality word alignments without parallel training data using static and contextualized embeddings. *arXiv preprint arXiv:2004.08728*, 2020.

R. L. Santos, C. Macdonald, and I. Ounis. Aggregated search result diversification. In *Conference on the Theory of Information Retrieval*, pages 250–261. Springer, 2011.

S. M. Sarwar, F. Moraes, J. Jiang, and J. Allan. Utility of missing concepts in query-biased summarization. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2056–2060, 2021.

P. Sen, D. Ganguly, M. Verma, and G. J. Jones. The curious case of ir explainability: Explaining document scores within and across ranking models. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2069–2072, 2020.

A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153. JMLR.org, 2017.

K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Y. Bengio and Y. LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

J. Singh and A. Anand. Posthoc interpretability of learning to rank models using secondary training data. In *Workshop on ExplainAble Recommendation and Search (EARS 2018) at SIGIR 2018*, 2018.

J. Singh and A. Anand. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 770–773, 2019.

J. Singh, M. Khosla, W. Zhenye, and A. Anand. Extracting per query valid explanations for blackbox learning-to-rank models. In *Proceedings of the 2021 ACM SIGIR International Conference on Theory of Information Retrieval*, page 203–210, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450386111. URL https://doi.org/10.1145/3471158.3472241.

J. Stacey, P. Minervini, H. Dubossarsky, and M. Rei. Logical reasoning with span predictions: Span-level logical atoms for interpretable and robust nli models. *arXiv preprint arXiv:2205.11432*, 2022.

M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3319–3328. JMLR.org, 2017.

N. Thakur, N. Reimers, A. Rücklé, A. Srivastava, and I. Gurevych. Beir: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/65b9eea6e1cc6bb9f0cd2a47751a186f-Paper-round2.pdf.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Fever: a large-scale dataset for fact extraction and verification. In *NAACL-HLT*, 2018.

J. Thorne, A. Vlachos, C. Christodoulopoulos, and A. Mittal. Generating token-level explanations for natural language inference. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 963–969, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017.

M. Verma and D. Ganguly. Lirme: Locally interpretable ranking model explanation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 1281–1284, 2019.

A. Williams, N. Nangia, and S. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, 2018.

Z. Wu, A. Naik, Z. X. Zhang, and L. Mou. Weakly supervised explainable phrasal reasoning with neural fuzzy logic. *arXiv preprint arXiv:2109.08927*, 2021.

F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199. ACM, 2008.

J. Xin, R. Tang, J. Lee, Y. Yu, and J. Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.204. URL https://www.aclweb.org/anthology/2020.acl-main.204.

J. Xu and W. B. Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems (TOIS)*, 18(1):79–112, 2000.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

A. Yates, R. Nogueira, and J. Lin. *Pretrained Transformers for Text Ranking: BERT and Beyond*, page 1154–1156. Association for Computing Machinery, New York, NY, USA, 2021. ISBN 9781450382977. URL https://doi.org/10.1145/3437963.3441667.

S. Yu, Z. Liu, C. Xiong, T. Feng, and Z. Liu. Few-shot conversational dense retrieval. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 829–838, 2021.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

J. Zhan, J. Mao, Y. Liu, M. Zhang, and S. Ma. *An Analysis of BERT in Document Ranking*, page 1941–1944. Association for Computing Machinery, New York, NY, USA, 2020. ISBN 9781450380164. URL https://doi.org/10.1145/3397271.3401325.

J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 15–20, 2018.

H. Zhuang, X. Wang, M. Bendersky, A. Grushetsky, Y. Wu, P. Mitrichev, E. Sterling, N. Bell, W. Ravina, and H. Qian. Interpretable ranking with generalized additive models. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pages 499–507, 2021.

L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=BJ5UeU9xx`.