Bachelor Thesis

Heidelberg University

Department of Computational Linguistics

Resolving Comparative Anaphora with and without Lexical Heads

JIN HUANG

Supervisor: Prof. Dr. Katja Markert

Co-Supervisor: Dr. Michael Herweg

March 2022

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386

Contents

1 Introduction **1**

1.1 Anaphora and Comparative Anaphora 1

1.2 Motivation 3

1.3 Research Hypotheses 3

2 Background Theory **5**

2.1 Distinction between Anaphora and Co-reference 5

2.2 Bridging 6

2.3 Comparative Anaphora with or without Lexical Heads 6

2.4 Neural Networks 7

2.4.1 Feed-Forward Neural Networks 9

2.4.2 Recurrent Neural Network 9

2.5 Transfer Learning 12

2.6 Embeddings 13

3 Related Works **16**

3.1 Other-Anaphora Resolution 16

3.2 Bridging Resolution 17

3.3 Comparative Anaphora Resolution 18

3.4 Pronominal Anaphora Resolution 19

4 Data **20**

4.1 ComAn Corpus 20
4.2 ISNotes Corpus 20
4.3 ARRAU Corpus 21
4.4 Complete Corpus 21
4.5 WithLex and WithoutLex 22
4.6 Data for Pre-Training 24

5 Approach **26**
5.1 Features 26
5.2 Model Architectures 28
5.2.1 Overall Model Setups 28
5.2.2 BiLSTM Model 29
5.2.3 ELMo Model 30
5.2.4 Pre-training and Transfer Learning 30

6 Experiments and Results **35**
6.1 Baselines 35
6.2 Experiment Setups and Data Splits 36
6.3 Evaluation Measure 37
6.4 Results 37
6.5 Error Analysis 40
6.5.1 Cosine Similarity Between Anaphor and Gold or Candidate Anteedents. 40
6.5.2 Qualitative Analysis of Predictions 40
6.6 Significance 44

7 Conclusion **46**

List of Figures **48**

List of Tables **49**

Abstract

Automatic anaphora resolution is an essential but challenging topic in the area of Natural Language Processing. It is crucial for natural language understanding systems such as discourse analysis, summarization, question answering, etc. Distinguishing from ordinary anaphora resolution, this work focuses on comparative anaphora resolution. "Comparative anaphora are referential noun phrases with non-pronominal heads which are introduced by lexical modifiers, such as *other, another, similar* or comparative adjectives, such as *better, greater*." [Klowersa 2021, p. 2]. The main task of this thesis is interpreting the relationship between the anaphor and its antecedent, and finding the correct antecedent of a referring comparative expression. The expressions of comparative anaphora are divided into two parts according to two different types — anaphor with or without lexical heads, i.e. the comparative anaphora corpus is divided into the WithLex and the WithoutLex corpus. This work applies neural methods to resolve comparative anaphora. One model is a BiLSTM model, which takes ELMo or GloVe embeddings as input of a BiLSTM network and learns the span representation of the anaphor and its candidates (abbreviated as the BiLSTM model). The models then rank the pairwise score of every anaphor and each of its potential antecedents to predict its correct antecedent. Besides, extra features are also added. Additionally, another simplified model is applied - The output of ELMo, i.e. ELMo embeddings, will be directly fed into an FFNN (abbreviated as the ELMo model). This model just omits the additional BiLSTM part of the BiLSTM model above, since ELMo itself contains a BiLSTM model. Because the comparative anaphora annotations are very limited, Transfer Learning is also applied. The ELMo model will be first pre-trained on the corpus I created for the pronoun resolution task. The model's parameters are optimized by training on the source task, pronominal anaphora resolution, and transferred to the target task, comparative anaphora resolution. The first hypothesis of this thesis is that **1. there are significant differences between comparative anaphors with and without lexical heads**. The second is that **2. resolving the antecedent of anaphor without lexical heads is extremely context-dependent**. The third is that **3. including extra features will improve the models' performance**. The fourth is that **4. pre-training on pronominal anaphora data might improve the model performance for the anaphora resolution task on WithoutLex**. The fifth is that **5. additionally concatenating Phrase-BERT embedding can improve the model performance**.

Zusammenfassung

Die automatische Auflösung von Anaphora ist ein wichtiges, aber schwieriges Thema im Bereich des NLPs. Sie ist von entscheidender Bedeutung für Systeme zum Verstehen natürlicher Sprache, wie z.B. Diskursanalysen, Zusammenfassungen usw. "Comparative anaphora are referential noun phrases with non-pronominal heads which are introduced by lexical modifiers, such as *other, another, similar* or comparative adjectives, such as *better, greater.*" [Klowersa 2021, p. 2]. Die Hauptaufgabe dieser Arbeit besteht darin, die Beziehung zwischen der Anapher und ihrem Antezedens zu interpretieren und das richtige Antezedens eines verweisenden komparativen Ausdrucks zu finden. Die Ausdrücke der komparativen Anapher werden in zwei Teile unterteilt, die sich nach zwei verschiedenen Typen unterscheiden - Anapher mit oder ohne lexikalische Köpfe, d.h. der Korpus von [Klowersa 2021] der komparativen Anapher wird in den WithLex und den WithoutLex Korpus unterteilt. In dieser Arbeit werden neuronale Methoden zur Auflösung der Anaphora eingesetzt. Ein Modell enthält ein BiLSTM-Netz, das ELMo- , GloVe-Einbettungen oder ihre Verkettung als Input nimmt und die Spannenrepräsentation der Anapher und ihres Antezedens erlernt. Die Modelle ordnen dann die paarweise Bewertung jeder Anapher und jedes ihrer potenziellen Antezedenzien, um ihr korrektes Antezedens vorherzusagen. Außerdem werden zusätzliche Merkmale hinzugefügt(abgekürzt als das BiLSTM-Modell). Zusätzlich wird ein weiteres vereinfachtes Modell angewandt - die ELMo-Einbettungen wird direkt in ein FFNN eingespeist(abgekürzt als das ELMo-Modell). Bei diesem Modell wird lediglich der zusätzliche BiLSTM-Teil des ersten Modells weggelassen, da ELMo-Modell selbst ein BiLSTM-Modell enthält. Da der Korpus von [Klowersa 2021] relative klein ist, wird auch Transfer Learning angewendet. Das ELMo-Modell wird erst auf dem Korpus vortrainiert, das ich für die Pronomenauflösung erstellt habe. Die Parameter des Modells werden durch Training in der Ausgangsaufgabe, der pronominalen Auflösung der Anaphora, optimiert und auf die Zielaufgabe, die komparative Auflösung der Anaphora, übertragen. Insgesamt gibt es fünf Hypothesen — **1.dass es signifikante Unterschiede zwischen komparativen Anaphern mit und ohne lexikalische Köpfe gibt**. **2. dass die Auflösung des Antezedens einer Anapher ohne lexikalische Köpfe extrem kontextabhängig ist**. **3. dass die Aufnahme zusätzlicher Merkmale die Leistung der Modelle verbessern wird**. **4. dass das Vortraining auf Pronominalanaphorendaten die Modellleistung für die Auflösung der Anaphora auf WithoutLex verbessern könnte**. **5. Zusätzlich verkettende Phrase-BERT-Einbettung die Modellleistung verbessern kann**.

Acknowledgement

In addition to my supervisors, I would also like to thank my colleagues Benjamin Beilharz and Michael Staniek for their advice on this work, and thanks to Victor Zimmermann for the proofreading.

Chapter 1

Introduction

First, I will introduce what is anaphora and comparative anaphora. Then I will explain the motivation and research hypotheses of this work.

1.1 Anaphora and Comparative Anaphora

Anaphora

"The etymology of the term *anaphora* goes back to Ancient Greek with "$\alpha\upsilon\alpha\phi o\rho\alpha$" being a compound word consisting of the separate words "$\alpha\upsilon\alpha$" - *back, upstream, back in an upward direction* and "$\phi o\rho\alpha$" - *the act of carrying* and denoted *the act of carrying back upstream.*" [Ruslan et al. 2000, p. 1].

[Halliday et al. 1976] made a classic definition for anaphora in general: "Anaphora is cohesion (presupposition) which points back to some previous item.". In other words, anaphora describes the dependence of two discourse elements. It is an expression whose interpretation strongly relies on another expression in the context. The referring term is an anaphor, which refers to its antecedent, the expression preceding the anaphor. Next, I will give some examples of different types of anaphora, all anaphors will be bold and all antecedents will be italic.

(1): *Willy* said **he** will be there.

In example (1), the pronoun phrase *he* is a pronominal anaphor referring to its antecedent, the proper noun *Willy.* Moreover, there are different types of anaphora, such as definite noun phrase anaphora, one-anaphora, etc. However, pronominal anaphora is the predominant type of anaphora, where the anaphor is a pronoun, including personal and possessive pronouns, reflexive and reciprocal pronouns, demonstrative pronouns, relative and interrogative pronouns, and indefinite pronouns.

Comparative Anaphora

Distinguishing from general anaphora resolution, this work focus on comparative anaphora resolution.

"Comparative anaphora are referential noun phrases with non-pronominal heads which are introduced by lexical modifiers, such as *other, another, similar* or comparative adjectives, such as *better, greater.*" [Klowersa 2021, p. 2]. The comparative anaphor is comparable to the antecedent it refers to. It can assume the following roles for its antecedent:

1. When the premodifier of the anaphor is *other, others, the other one, etc.*, this kind of comparative anaphors is mostly the set complements of its antecedent.
   (2): *My math teacher* went on vacation, but **the other teachers** worked as usual.

2. When the comparative anaphor is introduced by *similar, same, such, etc.*, it highlights its similarities with its antecedent.
   (3): He liked *my bag*, so he bought **a very similar bag**.

3. When the comparative anaphor is modified by comparative adjectives such as *better, bigger, more beautiful, etc.*
   (4): I enjoyed *this book*, but I borrowed **a more interesting one**.

However, the following cases in which the antecedent is standing in a structural relation to the anaphor are excluded:

1. In list-constructions as in "X and/or Y", where the antecedent and the anaphor are in the same co-ordinated noun phrase:
   (5): *Peppermint* and **other herbs** can relieve this disease.

2. In "other-than" construction:
   (6): He has **skills other than *programming***.

3. Idiomatic expressions:
   (7): *the other week, another day*

4. Reciprocal phrases:
   (8): *each other, one another*

Those cases are excluded from the comparative dataset used in this work because the antecedents of the other-anaphor in those structural constructions are structurally available. This work will focus on resolving more complex comparative anaphoric relations, instead of just using syntactic constraints to retrieve them.

1.2 Motivation

In text linguistics, *coherence* is an essential factor in determining the semantic meaning of natural language. Syntactic features, such as anaphora, contribute to coherence. In sentences, anaphora connect different syntactical elements and makes a text coherent. Thus, it is one of the keys to understanding the construction and maintenance of discourse.

Automatic anaphora resolution is an important but challenging topic. It is crucial for almost every natural language understanding system, such as discourse analysis, text summarization, machine translation, natural language generation, information extraction, question answering, etc. Comparative anaphora resolution is even more specific than anaphora resolution in general and there is not much work on it. Better understanding and resolving comparative anaphora is expected to improve full automatic reference resolution.

1.3 Research Hypotheses

This thesis aims to interpret the relationship between the two discourse elements and select the correct antecedent of a comparative anaphor. The possible candidate antecedents in the data set are all the noun phrases that occur before the anaphor in the *context* - the sentence where anaphor appears and the two sentences preceding it.

This work applies neural methods to resolve comparative anaphora. One model contains a BiLSTM network, which takes ELMo, GloVe, or their concatenation as input and learns the span representation of the anaphor and its antecedent. The model then ranks the pairwise score of every anaphor and each of its potential antecedents to predict its correct antecedent. Besides, positional, grammatical, and semantical features are also added. Additionally, another simplified model is applied - the ELMo model. The extracted and processed ELMo embeddings for the anaphor and the antecedent will be directly fed into a feed-forward neural network. This model just omits the additional BiLSTM part of the BiLSTM model above, since ELMo itself contains BiLSTM architecture already.

The first hypothesis of this thesis is that **there are significant differences between comparative anaphor with and without lexical heads**. In order to observe and confirm this, [Klowersa 2021]'s comparative anaphora dataset has been divided into two parts according to these two different comparative anaphor types.

Because of the extreme lack of lexical information, the second hypothesis that **resolving the antecedent of anaphor without lexical heads is extremely context-dependent**.

The third hypothesis is that **including extra features will improve the models' performance**. Thus, different extra features used are outlined in Section 5.1.

The fourth hypothesis is that **pre-training on pronominal anaphora data could improve the model performance for the comparative anaphora resolution task on Without-Lex**. Therefore, Transfer Learning is also applied. The BiLSTM model will be first pre-trained on the corpus I created for the pronoun resolution. The model's parameters are optimized by training on the source task, pronominal anaphora resolution, and transferred to the target task, comparative anaphora resolution.

The fifth hypothesis is that **additionally concatenating Phrase-BERT embedding can improve the model performance**.

Chapter 2

Background Theory

First, I will explain the linguistic definition of co-reference and bridging, as well as the difference between comparative anaphora with and without lexical heads. Second, there is an introduction to neural networks. Finally, two different word embeddings, GloVe and ELMo, as well as a phrase embedding method Phrase-BERT will be presented.

2.1 Distinction between Anaphora and Co-reference

Anaphora resolution is a task that interprets the relationship between the anaphor and its antecedent in a context and finds the correct antecedent.

Co-reference is another relevant definition which relates to anaphora. Co-references occur when two or more expressions in a text refer to the same entity in the real world. In other words, these expressions are *co-referential.*

In example (1), the anaphor *he* refers to its antecedent *Willy*, which both refer to the same entity in the real world. One or more co-referential references that refer to the same entity constitute a *co-reference chain.* Co-references existed in anaphora, cataphora, split antecedents, referring noun phrases, etc. Hence, the definition of anaphora and co-reference intersect but do not belong to each other. They are often used interchangeably perhaps because, among all anaphoric references, co-referential references are the most studied. That means the relationship between an anaphor and its antecedent is not always co-referential, like in example (2).

Besides, the co-reference chain of the gold antecedents is very relevant for the evaluation of the model training. By evaluation, if the predicted antecedent is the gold antecedent or the co-references of the gold antecedent in the context, we consider the prediction correct.

2.2 Bridging

In bridging anaphora [Clark 1975], the associated anaphor [Cruse 1980] and its antecedent are not co-referential.

(9): I was washing *my car* today when I noticed **one of the tires** was flat.

In example (9), there is a bridging link between the anaphor *one of the tires* and its antecedent *my car*. *My car* can not be interpreted correctly without the association with *one of the tires*. Here, *one of the tires* is a meronym in a part-of relationship with its holonym *my car*.

Inspired by the two-level RefLex annotation scheme by [Riester et al. 2017], bridging anaphora can be divided into two categories — *lexical* and *referential bridging*.

Lexical bridging describes the phenomenon that the two expressions are in lexical-semantic relations, such as meronymy or hyponymy, at the word or concept level, like example (9).

However, in the case of *referential bridging*, only noun phrases that are truly anaphoric and can not be interpreted without an antecedent will be considered. To be more precise, comparative anaphors are instances of referential bridging.

(10): The results of *the US* election are out. Joe Biden has become **the president**.

In example (10), the position of president is linked to a country or a company. Here, to understand what president *Joe Biden* is, *the president* is linked to its antecedent *the US*.

2.3 Comparative Anaphora with or without Lexical Heads

In this work, the expressions of comparative anaphora are divided into two parts according to two different types — anaphor with or without lexical heads.

1. Comparative anaphor with lexical heads

   In this case, the lexical heads are mostly a common or proper noun phrase. For example, *some other* ***friends****, the stronger* ***girl****.*

2. Comparative anaphor without lexical heads.

   On the contrary, in this case, the anaphor has no lexical head, for example, *others, the other, another*

Specifically, the anaphora without lexical heads are extracted as described below — If the anaphor's head is in the fixed list ["other", "others", "one", "ones", "another"], the anaphora will be extracted. If that's not the case, the anaphora will be extracted if the POS tag of the anaphor's head is not in the fixed list ['NOUN', 'PROPN', 'NUM'].

As we can see, anaphors with lexical heads have significantly more lexical information than anaphors without lexical heads. For example, the phrase *other men* indicates that the antecedent could be a person or people, while *others* indicates almost nothing without context. However, the latter often has more syntactic restriction and context information, compensating for the lack of an anaphor lexical head. The former is more similar to the co-reference in general, while the latter is more similar to pronominal anaphora since pronouns also mostly do not contain any lexical information, e.g. *it*. The similarity of pronouns and anaphors without lexical heads is that they both carry almost no information, the context does.

2.4 Neural Networks

Artificial Neural Networks (ANNs), usually called Neural Networks (NNs), provide a learning mechanism with applications to natural language problems. Neural networks are computational models inspired by how the biological neural networks process information in the human brain.

A Neural Network consists of neuron layers and each neuron is connected to another neuron with its weight. A vector of input features x_i is passed into an input layer and then a number of hidden layers. Each edge between two nodes is given a weight w_i. The initialized weights are updated during the training. The result z is computed by the weighted sum over all values from the incoming nodes. Finally, a bias value b is added.

$$z = \sum_{i=1}^{n} x_i w_i + b$$

The result z is then passed to a nonlinear activation function σ which normalizes the output of each neuron. This function decides if the information will be passed to the next neuron. The activation functions relevant to this work are listed in Table 2.1. After that, the other neurons receive the output a.

	$f(x)$	$f'(x)$	$f(x) \in$
sigmoid	$\frac{1}{1+e^{-x}}$	$\sigma(x)(1-\sigma(x))$	(0,1)
tanh	$\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	$1-tanh(x)^2$	(-1,1)
ReLU	max(0,x)	$=\begin{cases} 1, if x < 0 \\ 0, otherwise \end{cases}$ (2.1)	(0,∞)

Table 2.1: Activation functions referenced in this work with their derivative and range

$$a = \sigma(z)$$

By measuring the error, a loss function identifies how well a system models the data. For an output $\hat{y}$ and its gold label y, the loss function $L(\hat{y}, y)$, i.e. the difference between the model's predictions and the gold labels, should be minimized by training the neural network. Thus the weights will be updated so that the predicted output $\hat{y}$ of gets close to the gold label y.

Stochastic gradient descent (SGD) is typically used for this, which updates the parameters in every iteration by incrementing in the direction of the negative gradient of the loss function. To update the model's parameters, backpropagation [Werbos 1990] is used. After each forward pass, the weight matrix for each layer is updated through backward differentiation, i.e. backpropagation, which is based on the chain rule from differential calculus. Specifically, the partial derivative of the loss function with respect to each parameter of the network, i.e. the weights and biases, is calculated during backpropagation.

The update function for a parameter w can be defined as:

$$w_{new} = w_{old} - \frac{\partial L(\hat{y},y)}{\partial w_{old}} \cdot \gamma$$

where γ is the learning rate, or step size, which determines how much the parameters are changed per update. This is done iteratively for several epochs. An epoch is a whole cycle of the training data. SGD calculates the error and updates the model for each single sample. It is also called an online machine learning algorithm. *Batch Gradient Descent* updates the model at the end of each training epoch after all training samples have been processed and the errors are accumulated. However, a method in between, the *Mini-batch Gradient Descent*, is used in this work. It splits the training dataset into small batches that are used to calculate model error and update model parameters.

2.4.1 Feed-Forward Neural Networks

A feed-forward Neural Network (FFNN) is a neural network that contains no cycles or loops. It is one of the most straightforward neural networks. As its name says, the information in the FFNN only moves forward using backpropagation, from the input neurons, through the hidden neurons (if any hidden layer exists), and to the output neurons.

2.4.2 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are used to carry the input through time and generate information representations. It is often to feed the output (information representations) of an RNN into a feed-forward neural network which will then make a prediction.

The most significant difference from feed-forward neural networks is that the connections between neurons in RNNs form a directed graph along with a time series. Besides, RNNs have "memory" in their internal state, which is used to process variable-length sequential data as input and produce a fixed size vector to summarize it. RNNs consider not only the current input but also the previous information. The hidden state of the previous time step will be used to compute the output of the current time step so that the hidden state represents the context information based on previous input. The most basic form of RNN was proposed by [Elman 1990] (see Figure 2.1).

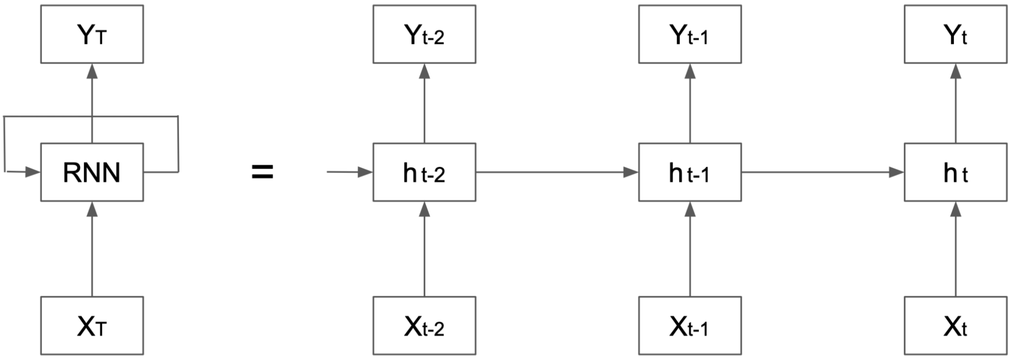

Figure 2.1: Recurrent Neural Network workflow

The rules for calculating the hidden state at time step t are as follows:

$$h_t = \sigma(W_h x_t + U_h h_{t-1} + b_i)$$

$$y_t = \sigma(W_y h_t + b_y)$$

These equations above, x_t denotes the input, h_t the current hidden state, h_{t-1}, the hidden state of previous time step $t-1$, and y_t the current output. W_h, U_h, W_y are weights metrics for inputs, hidden states as well as outputs, and b is the bias.

RNNs are often trained with gradient-based learning methods and Backpropagation Through Time (BPTT) to compute and store gradients. However, when the sequences are long, the long dependencies might cause the *vanishing gradients problem*. In this kind of approach, each of the weights will be updated proportionally to the partial derivative of the error function with respect to the current weight in each iteration. But sometimes the gradient is so tiny during the backpropagating that the weights can hardly be changed, and the model learns almost nothing.

Fortunately, a particular RNN model, the Long short-term memory (LSTM) model, can avoid the vanishing gradients problem.

Long Short-Term Memory Networks

Since RNNs usually suffer from short-term memory, the Long short-term memory (LSTM) model was proposed by [Hochreiter et al. 1997]. It is a kind of RNN that can learn long-term dependencies to avoid the vanishing gradients problem.

LSTMs contain mechanisms called gates that regulate the flow of information and learn which data in a sequence is important to "remember". A cell of a normal RNN will combine the previous hidden state and the current input, this combined vector will go through an activation function, and the output is the current hidden state. While a cell state of an LSTM, also called memory cell, contains three gates that regulate the information flow - forget gate, input gate, and output gate (see Figure 2.2).

The first gate is the *forget gate*, which decides which information needs to be "remembered" or can be "forgotten". The concatenation of the previous hidden state h_{t-1} and the current input x_t is passed through the sigmoid function. With output f_t is closer so 1, a larger portion of the information is kept, while id f_t is closer to 0, more information is lost, i.e. "forgotten".

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

The second gate is the *input gate*. The sigmoid function also receives the concatenation of the previous hidden state h_{t-1} and the current input state x_t. The values are transformed between 0 and 1.

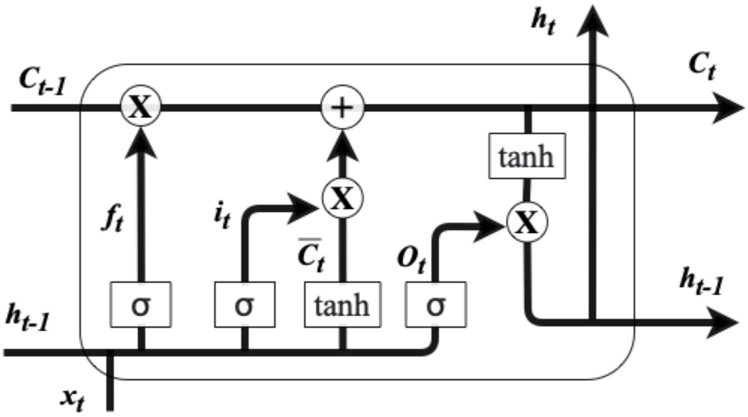

Figure 2.2: A BiLSTM cell

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i)$$

To regulate the network, the same concatenation is passed through the tanh function, and the output is $\bar{C}_t$.

$$\bar{C}_t = tanh(W_C[h_{t-1}, x_t] + b_C)$$

By adding the previous information the forget gate decides to keep $(f_t \cdot C_{t-1})$, and the new information the input gate decides to keep $(i_t \cdot \bar{C}_t)$, the new memory cell state C_t is updated.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t$$

The last gate, the *output gate* determines the value of the next hidden state that contains information on previous inputs. First, the concatenation of the previous hidden state and the current input will be passed into a sigmoid function to get the output of the output gate o_t.

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

Then the modified new cell state C_t will be passed through the tanh function. The value of the next hidden state is the multiplication of the tanh output and the output of the output gate o_t.

$$h_t = o_t \cdot tanh(C_t)$$

The new hidden state h_t is then the output y_t of the cell at time t and will be carried over to the next time step as the previous hidden state.

$$y_t = h_t$$

Theoretically, the cell state carries the chosen important information from earlier time steps throughout the processing of the whole sequence, thus reducing the vanishing gradients problem caused by the effect of short-term memory.

Bidirectional Long Short-Term Memory Networks

Bidirectional Long short-term memory neural networks (BiLSTM) by [Schuster et al. 1997] process information forward and backward by connecting two hidden layers (two LSTMs) of opposite directions to one output. Sequence processing tasks like anaphora resolution could benefit from accessing the sequence bidirectionally to acquire past and future information. The mainstream linguistic view is that interpretive semantics (of which anaphora is a part) happens after syntax, i.e. when the whole sentence is retrievable by the brain. This would motivate a bidirectional approach since a monodirectional LSTM does only have access to the previous part of the sentence.

Thus, a BiLSTM model and an ELMo model, which contains BiLSTM architecture, are used in this work. Figure 2.3 shows the workflow of a BiLSTM model.

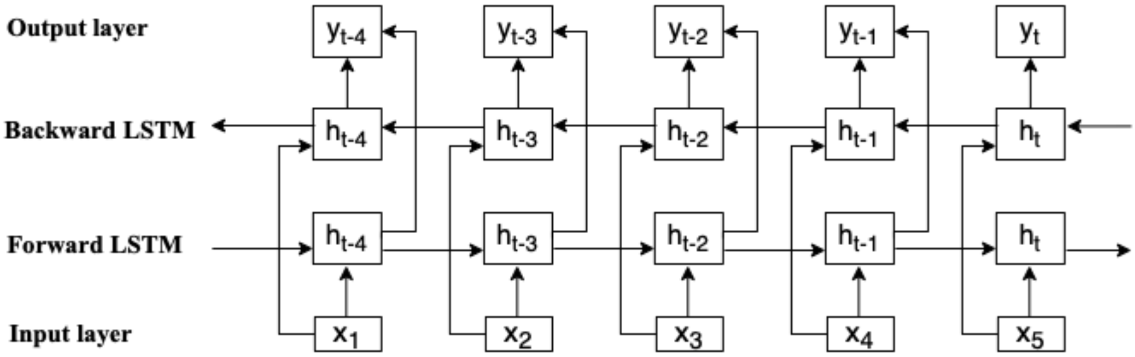

Figure 2.3: BiLSTM model workflow

2.5 Transfer Learning

Leveraging the knowledge learned by solving a problem and applying it to another related problem is called Transfer learning (TL). In Transfer Learning, a source model will first be trained on the source data for a source task. Then the learned features will be transferred to another target model, which will be trained on the target data for a target task.

It is a proper machine learning technique that can avoid overfitting when the data for a new domain is very limited, and sufficient existing data can be transferred to the real problem. For this reason, this technique is also employed in this work.

2.6 Embeddings

GloVe

Two different embeddings are involved in this work. One is Global Vectors (GloVe) by [Pennington et al. 2014], a model for distributed word representation. With this learning algorithm, global word co-occurrence statistics are leveraged to map words into a meaningful vector space in which the distance between word vectors is correlated with their semantic similarity. But this embedding has the disadvantage that homographs, i.e. words with the same spelling but different meanings, have the same vector. However, Embeddings from Language Model(ELMo) overcame this problem.

ELMo

Embeddings from Language Model (ELMo) by [Peters et al. 2018] is a deep contextualized word representation that uses a pre-trained language model trained on a massive dataset to predict the next word in a sequence of words.

Firstly, a character-level convolutional neural network is applied for producing the initial word embeddings. Since ELMo uses a multi-layer bidirectional language model (biLM), the initial embedding is fed into the first BiLSTM layer of the biLM. Then, the internal states of both forward and backward passes get concatenated and form an intermediate word vector. Thus, this intermediate word vector at that word not only represents what the word means but also how this word is used in the context in both directions.

Additionally, two BiLSTM layers are stacked together and form a multilayer BiLSTM. However, in contrast to typical multilayer BiLSTM, in ELMo, the resulting vectors are concatenated only after the input is processed respectively in each BiLSTM layer in each direction. More specifically, the intermediate word vector generated by the first BiLSTM layer below is then fed into the next BiLSTM layer. In this way, the internal state can be processed further in the next BiLSTM layer. The first layer can capture part of speech, while the second layer learns word-sense representations and represent more abstract semantics such as topics and sentiment.

The final contextualized ELMo representation is the weighted summation of the intermediate word vectors of different layers and the initial embedding. Unlike Glove, homographs have different ELMo representations in different contexts.

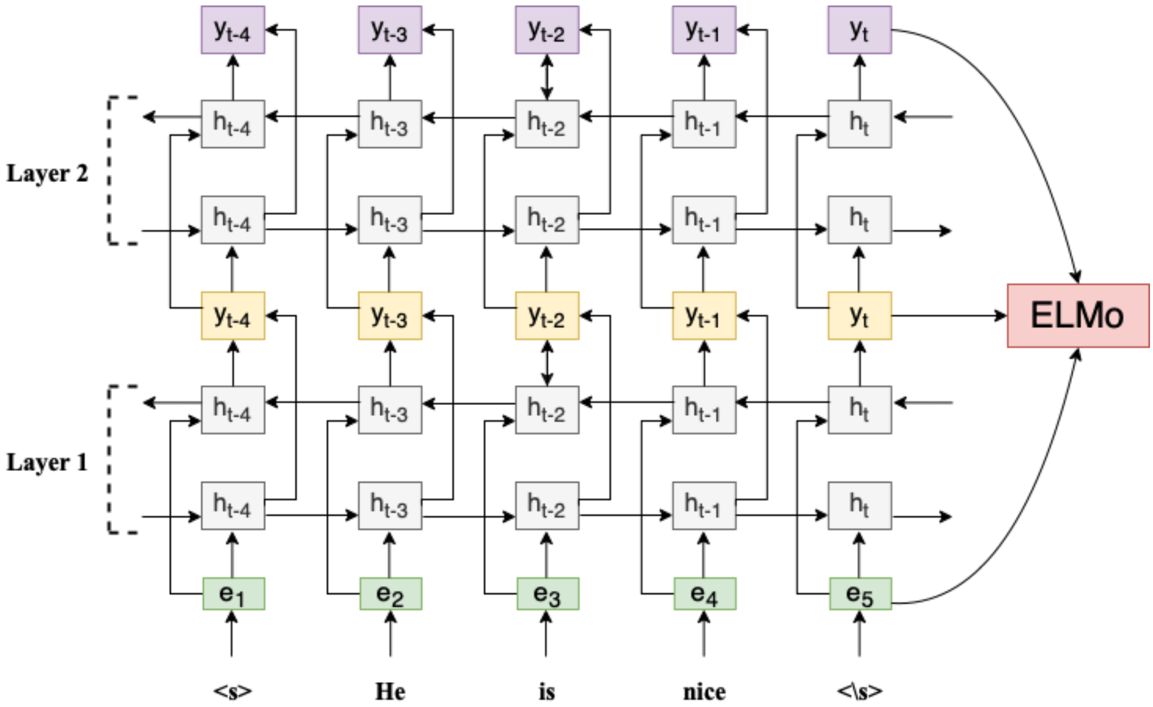

Figure 2.4: ELMo uses a multi-layer biLM

Phrase-BERT

Phrase-BERT [Wang et al. 2021] induces high-quality phrase embeddings by fine-tuning BERT [Devlin et al. 2019] with two contrastive objectives on two different datasets.

The authors use BERT for phrase embedding by passing the tokens of the phrase to the BERT model to calculate a phrase representation by averaging the final-layer, i.e. the token-level vectors, from BERT during the fine-tuning.

Their first fine-tuning objective encourages BERT to capture semantic relatedness between phrases. They create a data set of lexically diverse phrasal paraphrases. They pass every phrase p through the GPT2-based diverse paraphrasing model [Krishna et al. 2020] to create a positive example p^+. Then, they randomly select a non-stopword from p and replace it with a random token. After that, they also feed the corrupted phrase into the paraphrasing model to get a negative example p^-. In This way, the lexical overlap is eliminated, but the distorted meaning is retained.

The second fine-tuning objective encourages the model to include contextual information into phrase embeddings. They create a phrase-in-context dataset by extracting phrases and keeping the positive context c^+ where p appears, p is then replaced with a MASK token. They also randomly sample a context c^- for calculating the loss.

Both datasets are fine-tuned with BERT using the same contrastive objective, as with Sentence-BERT [Reimers et al. 2019]. For the first dataset, they encourage the model to generate similar embeddings for p and $p+$ while pushing the embeddings for p and $p-$ far apart.

By mean-pooling BERT's token-level representations, they embed (p, p^+, p^-) to create three new vectors ($\mathbf{p}$, $\mathbf{p}^+$, $\mathbf{p}^-$) for each phrase. After that, they calculate the triplet loss, where $||\cdot||$ denotes the $L2$ norm and ϵ is a margin ($\epsilon = 1$):

$$J_1 = max(0, \epsilon - ||\mathbf{p} - \mathbf{p}^-|| + ||\mathbf{p} - \mathbf{p}^+||)$$

And for the second dataset, they encode (p, c^+, c^-) with Phrase-BERT for each data instance ($\mathbf{p}$, $\mathbf{c}^+$, $\mathbf{c}^-$) and then calculate the triplet loss:

$$J_2 = max(0, \epsilon - ||\mathbf{p} - \mathbf{c}^-|| + ||\mathbf{p} - \mathbf{c}^+||)$$

Chapter 3

Related Works

In this chapter, some of the few current research works on the task of comparative anaphora resolution, and works involving comparative anaphora resolution, i.e. other-anaphora resolution and bridging resolution, will be presented. Finally, a few works about pronominal anaphora resolution are also included, since it is related to the pre-training of this work.

3.1 Other-Anaphora Resolution

An other-anaphora resolution model is approached by [Modjeska et al. 2003], employing a Naive Bayes classifier that considers morpho-syntactic, recency, string matching, and semantic features. Besides, they come up with a better way than WordNet to integrate semantic knowledge by counting web frequency from lexico-syntactic patterns specific to other-anaphora (web feature). It is proposed that an implicit relation between anaphor and antecedent in the same context may also be explicitly expressed in the others. The F1-score of the Naive Bayes classifier has increased from 45.5% to 56.9% with the web feature.

In another other-anaphora resolution work, [Modjeska 2003] applies two resolution algorithms — LEX and SAL. LEX is based on heuristics, which leverages lexical information from WordNet, pattern matching, recency, and heuristics for Named Entity (NE) antecedents. SAL is based on Centering Theory by [Grosz et al. 1995], which utilizes the grammatical salience between antecedent and other-anaphor. In other words, it emphasizes the grammatical function of a Noun Phrase (NP). Although the authors conclude that lexical and semantic information has more impact than grammatical information on resolving other-anaphora, combining semantic and grammatical resources provided the best results. They also use a machine learning model based on the Naive Bayes classifier. It follows the approach outlined above in [Modjeska et al. 2003]. A hybrid system is then applied, which contains both a probabilistic model and a set of informed heuristics and back-off procedures. This achieves the best success rate for 54.25%.

An other-anaphora resolution in bio-medical texts is proposed by [B. Chen et al. 2008]. They adopt the learning-based model used by [Soon et al. 2001] and [Ng et al. 2002] to resolve identity-anaphoras. They not only integrate features that rely on shallow lexical, position, and semantic information about an anaphor and a candidate antecedent but also leverage auto-mined pattern features to disclose the "part-whole" relation between the two mentions. Compared to the manually designed pattern, the auto-mined pattern improves the accuracy from 47.5% to 56.5%.

3.2 Bridging Resolution

As described in Section 2.2, comparative anaphora resolution is a subtask of bridging resolution.

[Rösiger et al. 2018] present two models for bridging resolution: a rule-based approach by [Hou et al. 2014] and a learning-based approach. By the reimplementation of the rule-based system of [Hou et al. 2014], they include eight hand-crafted linguistic rules focusing on referential bridging. Because the model of [Hou et al. 2014] performs poorly on the ARRAU corpus, they assume that this system suggests mostly referential bridging pairs, while ARRAU has mainly lexical bridging pairs. Their learning-based approach is the model by [Hou et al. 2014] plus new rules to capture lexical bridging. The new rules have been developed on the RST domain of the ARRAU corpus. This is described in another paper of [Rösiger 2018] (an extension paper of [Rösiger et al. 2018]) in details. Some of the new rules are specifically for the comparative anaphora case — For every anaphor, the system searches for the closest markable of the anaphor, which belongs to the same category as the anaphor. Then it will check if the head of this markable matches the anaphor's head in the last seven sentences. Otherwise, it searches for a markable in the context (the same and previous sentence in front of the anaphor) which belongs to the same category as the anaphor. If this search is also unsuccessful, it searches in the context for either a markable with the same head as the anaphor or the markable, which is a WordNet synonym of the anaphor. Afterward, an SVM classifier is used. In the full bridging resolution task, the learning-based approach achieves a precision of 57.7, a recall of 10.1, and an F1 score of 17.2. They get a higher precision compared to [Hou et al. 2014], but a lower recall, resulting in a lower F1 score.

3.3 Comparative Anaphora Resolution

In [Zimmermann 2019]'s work, a neural network architecture is applied to comparative anaphora resolution for the first time. He creates a corpus of 512 in-text comparative anaphora samples and automatically collects a corpus of 3825 co-reference mention pairs from the OntoNotes corpus for pre-training. He applies a BiLSTM model on his comparative anaphora corpus. In addition, he pre-trains the model on his co-reference corpus, then applies it to his comparative anaphora corpus. The transfer learning model achieves a micro-averaged result of 32.81%, which is 3.12% higher than without pre-training.

[Klowersa 2021] applies several neural methods to resolve comparative anaphora resolution. The main model is a BiLSTM model inspired by [Zhang et al. 2019], which is used to encode the anaphors and candidates spans into their contextual representations.

For each anaphor, the model ranks the anaphor-candidate pairs according to a pairwise score that decides whether there is an anaphoric relation between the anaphor and the candidate. The model will select the candidate with the best score as a true antecedent. It performs binary classification similarly to a mention-pair model that includes a ranking component. Except for the contextual knowledge that the BiLSTM encodes, the model also includes the external knowledge from Knowledge Graph, WordNet, and other features acquired from the corpus. All eight features (distance, substring, length, grammatical role, surface form, synonym, hypernym, and knowledge graph) were added to the model after encoding the text with BiLSTM and the span representation extraction.

Another alternative model is called the mention selection model, which is also based on a BiLSTM model. This model takes the entire sequence with the anaphor and all candidates as a sample. For every sample, the anaphor has a unique integer, and the integers assigned to the candidates are obtained by counting backward, starting with the closest candidate to the anaphor. Each such integer vector is added up together, acts like highlighting, and passes through the pre-trained embedding layer. The same is done for the anaphor span. Then, the features are added to the output of the BiLSTM model. Finally, A softmax function calculates the probability distribution of all candidates and selects the candidate with the highest score is as the true antecedent.

In addition, different transfer learning methods are applied. One of them is to pre-train the model on co-reference data and then transfer the learned weights to the main task on comparative anaphora resolution. Another is that the pre-trained language model BERT is applied in fine-tuning and a feature-based approach. Moreover, they applied the third transfer learning by combining the two above described Transfer Learning methods. On the comparative anaphora corpus, the BiLSTM model shows a success rate of 20.48%. The model which is pre-trained on the co-reference data archives a success rate of 31.25%. And the combined Transfer Learning reaches the highest success rate of 36.30%

3.4 Pronominal Anaphora Resolution

Pronominal anaphora resolution, also called pronoun resolution, is one of the most common tasks of anaphora resolution. A zero pronoun resolution model by [Liu et al. 2016] generates and utilizes large-scale pseudo training data for zero pronoun resolution. Zero pronouns refer to language components that have been omitted because of coherence. The noun phrase(s) a zero pronoun co-refers to in the preceding text is its antecedent.

They first generate training data automatically. For their zero pronoun resolution task, they use a very similar model as the attention-based neural network model by [Hermann et al. 2015] for cloze-style reading comprehension tasks. [Liu et al. 2016] first pre-train the model on the generated large-scale training data. After obtaining the best model, they train the model on the zero pronoun resolution task-specific training data. With a 55.3% F-score, the model outperforms the state-of-the-art system of Chinese zero pronoun resolution of [C. Chen et al. 2016] by 3.1%.

However, there are not a lot of articles about pronoun resolution in particular, because most people focused on the coreference-resolution task which includes pronoun resolution.

Chapter 4

Data

The total comparative anaphora corpus in this work is drawn from the combined dataset created by [Klowersa 2021], which consists of partial data from three different anaphora corpora. This will be explained in detail below. In addition, a corpus I created for pre-training will be presented.

4.1 ComAn Corpus

The comparative anaphora dataset created by [Zimmermann 2019] is a manually annotated dataset containing 512 samples from the OntoNotes corpus (OntoNotes comprises various genres of text, i.e. newswire, broadcast news, broadcast conversation, telephone conversation, web data, and English pivot text (Old Testament and New Testament text)). We will abbreviate it as CompAn corpus from now on. The comparative anaphora in this corpus are discovered by finding comparative adjectives such as "bigger", "better", modifiers like "more", "less" combined with another adjective like "more interesting", or from a fixed list of specific lexical modifiers e.g. "other", "another", "similar", etc.

[Klowersa 2021] deletes several unclear annotations and extracted 507 remaining samples.

4.2 ISNotes Corpus

[Klowersa 2021] selected 155 more comparative anaphora from ISNotes, which is created by [Markert et al. 2012] for classification of fine-grained information status.

[Markert et al. 2012] extract 11,000 annotated noun phrases from 50 texts from the WSJ domain of the OntoNotes corpus and add a fine-grained information status layer to this extraction. The comparative anaphora can be easily extracted from ISNotes because three anaphora types are annotated: co-reference, bridging anaphora, and comparative anaphora.

4.3 ARRAU Corpus

The last data source is ARRAU developed by [Uryupina et al. 2020] over 10 years. It is an anaphorically annotated corpus that contains annotation of plural anaphora, anaphora to abstract objects, and ambiguous anaphoric expressions. This collection includes bridging references and co-reference. It also provides rich linguistics information such as (non-)referentiality status, anaphoric ambiguity as well as morphosyntactic and semantic mention.

The corpus contains different genres of data, including a substantial amount of news text covered by the RST domain, the task-oriented dialogues from TRAINS domain extracted from TRAINS-93 corpus, the complete collection of spoken narrative in the Pear Stories domain from PEAR sub-corpus, and documents from the medical and art history genres from GNOME sub-corpus.

Comparative anaphora can be selected from ARRAU because every referring NP can also be marked as related to a previously occurring mention. As part of the bridging reference annotation, the annotation is however restricted to "other NPs". Those 'other anaphora' annotations are only available in RST, thus the samples only come from news text.

227 samples of "other anaphora" from the RST domain were selected by [Klowersa 2021].

4.4 Complete Corpus

After the removal of duplications and anaphora where the antecedent is syntactically available, there are 754 samples in total. The CompAn corpus hast the most samples, i.e. 498 samples, followed by 134 samples from ISNotes and 122 from ARRAU. (see Table 4.1)

Table 4.2 shows the numbers of anaphors per different data genres. The largest proportion of the total anaphors is from news articles from WSJ (from OntoNotes and Penn Treebank), followed by pivot text data, i.e. New Testament from OntoNotes Corpus.

	ARRAU	ISNotes	CompAn	
all	227	207	507	
w/o syntactic samples	141	155	507	
after removing duplicates	122	134	498	
total				754

Table 4.1: Number of anaphors in the total comparative anaphora corpus

Genre	Nr.of anaphors	ratio
Newswire	491	65%
Pivot Text	133	17%
Web	58	8%
Broadcast news	52	7%
Broadcast conversation	13	2%
Telephone onversation	7	1%

Table 4.2: Genre of the anaphors in the total comparative anaphora corpus

4.5 WithLex and WithoutLex

To study these two phenomena separately, I divided the complete comparative anaphora corpus into two parts according to two different comparative anaphor types — anaphor with or without lexical heads. About 74% of the anaphors are with lexical heads, while the remaining 26% are without lexical information (see Table 4.3). For convenience, I will refer to the corpus containing the anaphora with lexical heads as WithLex for short, and the other one as WithoutLex.

Corpus	Nr.of anaphors
With lexical info	560
Without lexical info	194
Total	754

Table 4.3: Distribution of the split corpus

Table 4.4 presents an interesting comparison of the quantitative statistics of the WithLex and WithoutLex corpus. The average number of potential candidates of WithoutLex is 14.5, which is slightly less than WithLex (16.7), and the average *context* length of WithoutLex is 3.7 (the context length is the number of words in the context), which is also shorter than WithLex (5.2). The distance between the anaphor and the gold antecedent is defined by the number of candidates between the anaphor and the gold antecedent in a context. As expected, the gold antecedent and its co-references in WithoutLex are nearer to its anaphors by 1.5 candidates than in WithLex. This indicates a natural linguistic phenomenon, that when the anaphor has no lexical head, its antecedent has to be as close as possible. Otherwise, the reference between those two expressions might be ambiguous or even uninterpretable, while the information of this kind of anaphor to identify antecedent is already minimal.

	WithLex		WithoutLex		Total	
	incl. coref.	excl. coref.	incl. coref.	excl. coref.	incl. coref.	excl. coref.
Nr. cand.	16.7	7.8	14.5	6.9	16.0	7.5
Distance	5.2		3.7		4.8	
Context len	73.7		60.2		70.3	

Table 4.4: Quantitative statistics of data

Table 4.5 indicates that there are 38.39% of the anaphors in WithLex and 55.67% of the anaphors in WithoutLex are *intrasentential*, whose antecedent is in the same sentence as itself. To also include the *intersentential* anaphors, whose antecedent is in a different preceding sentence from that of itself, the sentence window for the *context* for an anaphor is set to 3 (n=2). Table 4.5 indicates that this sentence window covers 94.16% antecedents in the total corpus. Especially, this ratio of WithoutLex is higher than WithLex, i.e only 1.55% gold antecedents are not available in the sentence window in WithoutLex.

N	WithLex	WithoutLex	Total
0	38.39%	55.67%	42.84%
1	88.93%	96.39%	90.85%
2	**92.68%**	**98.45%**	**94.16%**
3	94.64%	98.97%	95.76%
4	96.61%	98.97%	97.21%
5	96.61%	99.48%	97.35%

Table 4.5: Ratio of gold antecedents in same and last n sentences preceding the anaphor

Another informative frequency is displayed in Table 4.6. 57.33% of anaphors in WithoutLex have the gold antecedents as a subject, which is 20.63% more than in WithLex.

	Number	Ratio
WithLex	205	36.70%
WithoutLex	111	57.33%
Total	355	42.18%

Table 4.6: Frequency of the antecedents being a subject

Table 4.7 shows the distribution of the linguistic types of the gold and candidate antecedents, including linguistic expressions like demonstrative, definite, indefinite NPs or NPs without a determiner, pronouns, and proper names. In WithLex, most of the gold antecedents are pronominal NPs for 26%, followed by proper name NPs for 21%, while most of the gold antecedents in WithoutLex are also pronominal NPs for 53%, followed by proper name NPs and possessive NPs both for 14%.

	WithLex		WithoutLex		total	
	all cand.	gold antec.	all cand.	gold antec.	all cand.	gold antec.
definite NP	13%	11%	12%	4%	13%	9%
indefinite NP	8%	4%	8%	2%	8%	3%
demonstrative NP	2%	2%	3%	2%	2%	2%
proper name NP	18%	21%	14%	14%	17%	18%
pronominal NP	10%	**26%**	17%	**53%**	11%	37%
(poss.) pron. NP	8%	13%	8%	14%	8%	13%
no determiner	42%	23%	38%	11%	41%	18%

Table 4.7: Distribution of linguistic types of the gold and candidate antecedents

These statistics confirm that anaphor with and without lexical heads are different linguistic phenomena that deserve to be treated differently in training. Besides, this corpus is very small, so it is necessary to find a more extensive corpus for a similar task to pre-train the model beforehand.

4.6 Data for Pre-Training

The data used for the pre-training on the pronominal anaphora resolution task were extracted by me from the training data portion of the OntoNotes corpus. The duplicate samples from this corpus and the above total comparative anaphora corpus were removed from this corpus. All the pronominal anaphora were extracted, i.e. all anaphor whose head's POS-Tag is PRP or $PRP.

For every pronominal anaphor, all the mentions, i.e. the NPs, in the *context* precede the anaphor, which has any co-reference, were selected as the candidate antecedents. The gold antecedents are extracted from the selected candidates in the same co-reference chain as the anaphor.

This corpus for pre-training is much larger than WithoutLex, it contains 54113 pronouns (see

Table 4.8). From Table 4.8 we can see the similarity between these two types of anaphor, which is reflected in the fact that their average context length and the average number of candidates are very close.

	Pre-train	WithoutLex
Avg. cand. num	6.2	6.9
Avg. context len	54.4	60.2
Num of pronouns	54113	194

Table 4.8: Quantitative statistics of pre-train data and WithoutLex

However, from Table 4.9 we can see that the distances of these two different types of anaphor from their respective antecedent vary. In WithoutLex, most antecedents are in the same sentence as the anaphor, while in the pre-train data most antecedents are in the sentences in front of the anaphor sentence. But for both corpora, more than 98% of the antecedents are in the *context*.

N	Pre-train	WithoutLex
0	10.16%	55.67%
1	54.06%	96.39%
2	**100%**	**98.45%**
3	100%	98.97%
4	100%	98.97%
5	100%	99.48%

Table 4.9: Ratio of gold antecedents in same and last n sentences preceding the anaphor in the pre-train data

Chapter 5

Approach

Below are all the described feature types that I used in this work. Thereafter, different neural network architectures and methodologies used in this thesis to resolve comparative anaphora are explained in detail.

5.1 Features

Six different features used by [Klowersa 2021] are leveraged in the model in this work for the training. Except utilizing corpus data, WordNet [Fellbaum 2000] is also used for extracting the semantic features, as the semantic types of the comparative anaphor and its antecedent are often synonyms or hypernym of each other. Following is a detailed description of the individual features. A full list of all features can be found in Table 5.1.

Distance feature:

The distance feature is obtained by counting the number of the potential antecedents, i.e NPs, between each anaphor-candidate pair. These numbers are divided into nine buckets [0,1,2,3,4,5-7,8-15,16-31,32-50] and this feature vector with the length of nine is encoded as a one-hot vector, i.e. if there is only one candidate between the anaphor and the candidate, the feature vector will be [0, 1, 0, 0, 0, 0, 0, 0, 0].

Substring feature:

The substring feature is calculated by counting the number of overlapping words between the lemmatized anaphor strings and the lemmatized potential antecedent strings. This number is normalized by dividing by the amount of all the candidates.

type	description	value
positional	Number of candidates between the anaphor-candidate pair	[0,1,2,3,4,5-7,8-15,16- 31,32+]
match	Number of overlapping words between the lemmatized anaphor and candidate strings (Normalized by the number of candidates)	continuous float value
grammatical	Grammatical role	"csubj", "csubjpass", "dative", "nsubjpass", "obj", “iobj", “oprd", “dobj”, “pobj”, “nsubj"
grammatical	definiteness	“definite”, “indefinite”, “demonstrative”
semantic	Is the head noun of the anaphor a synonym of the head noun of the candidate?	yes, no
semantic	Is if the head noun of the anaphor a hypernym of the head noun of the candidate?	yes, no

Table 5.1: Features used for training

Grammatical role feature:

This feature encodes the grammatical role of the anaphor and its candidates into a one-hot vector. The grammatical roles are acquired by a dependency parse of the context from spaCy [Honnibal et al. 2017] and each of the grammatical role can be one of the following values: [" csubj"," csubjpass"," dative"," dobj"," nsubj"," nsubjpass"," obj"," pobj"," iobj"," oprd"].

Concatenating the feature vector of the anaphor and the candidate yields the final feature vector for each anaphor-candidate pair.

Definiteness feature:

The definiteness feature can be *definite*, *indefinite*, or *demonstrative*. Here the case *no determiner* is ignored. The entire lists of possible values of each feature are given by [Klowersa 2021, p. 36]:

definite = ["the", "all", "both", "either", "neither", "no", "none"]

indefinite = ["a", "an", "each", "every", "some", "any," "few", "several", "many", "much", "little", "most", “more", "fewer", "less"]

demonstrative = [’this’, ’these’, ’that’, ‘those’]

The final feature vector for one anaphor-candidate pair is the concatenation of the feature vector of the anaphor and the candidate.

Synonym feature:
The synonym feature determines if the head noun of the anaphor is a synonym of the head noun of the candidate. The value will be one or zero. The head of these two noun phrases will be extracted by the dependency tree from spaCy, and the synonyms are retrieved from WordNet using NLTK [Loper et al. 2002]. This feature should only improve the performance of the model on WithLex but not on WithoutLex, since the anaphors in WithoutLex has no lexical heads.

Hypernym feature:
Hypernym feature checks whether the head noun of the anaphor is a hypernym of the head noun of the candidate. Likewise, the synonyms of the anaphor's heads are retrieved from WordNet and the value of this feature can be one or zero. This feature should also only improve the performance of the model on WithLex for the same reason above.

5.2 Model Architectures

5.2.1 Overall Model Setups

This overall model setup is used in all models of this work, i.e. the BiLSTM model, the ELMo model, and the pre-train model (training the ELMo model on the pronoun anaphora corpus).

For each anaphor a, its context c consists of the sentence where the anaphor appears and the two previous sentences of the anaphor sentence. Besides, there is a gold antecedent g, the co-references R of g, and a set of potential antecedents p_i denoted as P, which consists of all the noun phrases preceding the anaphor in c.

For each anaphor a, the model pairs it with each of its potential antecedents p_i respectively. Each such anaphor-candidate pair is one training sample for the model. Finally, the antecedent is predicted by selecting the highest-scoring anaphor-candidate pair of the anaphor.

The model's learning objective is to predict whether there exists an anaphoric relationship between the anaphor-candidate pair, i.e. if the predicted candidate p_i is the gold antecedent g or belongs to the gold antecedent's co-references set R, thus it performs a binary classification.

There is an overview of all the information for an anaphor I extracted from the corpus:

***anaphor**: others*

***context**: people will love only themselves and money. they will be proud and boast about themselves. they will abuse others with insult.*

***candidates**: people, only themselves and money, themselves, money, they, themselves, they*

***gold**: they*

***gold co-references**: people, themselves, they, themselves, they*

The span representation of the anaphor and the antecedent, i.e. $\hat{a}$ and $\hat{p}$, will be obtained in different ways in the BiLSTM model and the ELMo model, which will be explained later. The extracted representations $\hat{a}$ and $\hat{p}$ are concatenated. This pairwise span representation is then concatenated with the pairwise feature vectors f_i or the pairwise Phrase-BERT span representation, thus the knowledge representations or Phrase-BERT embeddings are utilized.

$$u_i = [\hat{a}, \hat{p}]$$

$$f_i = [f_1, f_2, ..., f_n]$$

This combined representation will be passed through a feed-forward neural network consisting of four linear layers with ReLU activation function.

$$o_i = FFNN([u_i, f_i])$$

The last layer is a fully-connected layer with a Sigmoid activation function, which calculates the final anaphoric score for each anaphor-candidate pair sample. For each anaphor, the candidate with the highest score will be predicted as the true antecedent.

$$y_i = \sigma(o_i)$$

5.2.2 BiLSTM Model

The first comparative anaphora resolution model of this work is a BiLSTM model inspired by [Klowersa 2021], which is inspired by [Zhang et al. 2019]. The span of each anaphor and its potential antecedents are encoded into span representations using a BiLSTM model. The hidden layer of the BiLSTM model has the size of 256.

Figure 5.1 illustrates the overall architecture, which explains the steps of this model. For each sample, i.e. an anaphor-candidate pair, every token in the context sequence c will be embedded, resulting in respective word embeddings. GloVe and ELMo embeddings are used respectively and in combination in different experiments. This embedded context sequence will then be fed into a BiLSTM layer.

The previous time step's hidden state h_{t-1} and the current pre-computed word embedding input e_t at time step t will be passed through the BiLSTM at each time step t:

$$h_t = BiLSTM([h_{t-1}, e_t])$$

After that, the span representation of the anaphor a and the candidate p_i will be extracted from the output hidden state of the BiLSTM for each input time step by index, respectively. The span representations provide important contextual information because the current cell state of the span stores contextual information before and after the span. Afterward, the span representations will be processed as described in Section 5.2.1.

5.2.3 ELMo Model

To examine whether a less complex model would perform better, the second model, i.e. the ELMo model, a simplified version of the BiLSTM model omitting the BiLSTM network after the embedding layer, such that no further contextualization is done since ELMo itself already utilizes BiLSTM architecture to encode contextually (see Section 2.6). That is, the span representations of the anaphors and their candidates will be extracted from the ELMo embeddings of the context. And these extracted span representations will then be processed as described in Section 5.2.1. Figure 5.2 illustrates the overall architecture.

5.2.4 Pre-training and Transfer Learning

Pre-training - Pronominal Anaphora Resolution

This work also leveraged Transfer Learning because the comparative anaphora corpus is very limited. Due to the similarity of pronominal anaphor and anaphor without lexical heads, I assume that pronoun resolution could be a proper pre-training to improve the model performance on WithoutLex. The similarity of pronouns and anaphors without lexical heads is that they both carry almost no information, the context does. Besides, these two phenomena also have something in common statistically (see Table 4.9), i.e. they have similar average candidates number and average context length. However, there are also differences between these two phenomena. For example, pronominal anaphora sometimes needs gender agreement but comparative anaphora without lexical heads does not. Also, the antecedents are mostly closer to the comparative anaphors without lexical heads than to the pronominal anaphors.

Therefore there's a good chance that Transfer Learning could improve the model performance. The corpus I created for the pronoun resolution is significantly larger than WithoutLex, thus the model can learn more complex and richer word embeddings. The ELMo model (Figure 5.2) is then pre-trained on the source corpus and the parameters are optimized by training on the source task, pronominal anaphora resolution, and then transferred to the target task, comparative anaphora resolution on WithoutLex.

Transfer Learning

Two different transfer learning methods are used, the only difference between them is whether the layers of the pre-trained model are frozen or not.

Transfer Learning 1 - Freeze the Layers

In the first transfer learning method, all layers in the pre-trained model are frozen, so their weights are not updated during the training on WithoutLex. The last fully-connected linear layer in the feed-forward neural network and the output layer with Sigmoid activation are deleted. Then, a new trainable linear layer and a new output layer with Sigmoid activation are added, i.e. they are unfrozen. In this way, the information learned during the pre-training on the source task will be preserved and leveraged on the target task. This setup is inspired by [Klowersa 2021]. Figure 5.3 illustrates the workflow.

Transfer Learning 2 - Fine-tune the Parameters

The second transfer learning method is done without freezing any layers, i.e. directly using the pre-trained model, so that the error can be backpropagated through the network during the training on WithoutLex. Thus, all model parameters are fine-tuned for the target task.

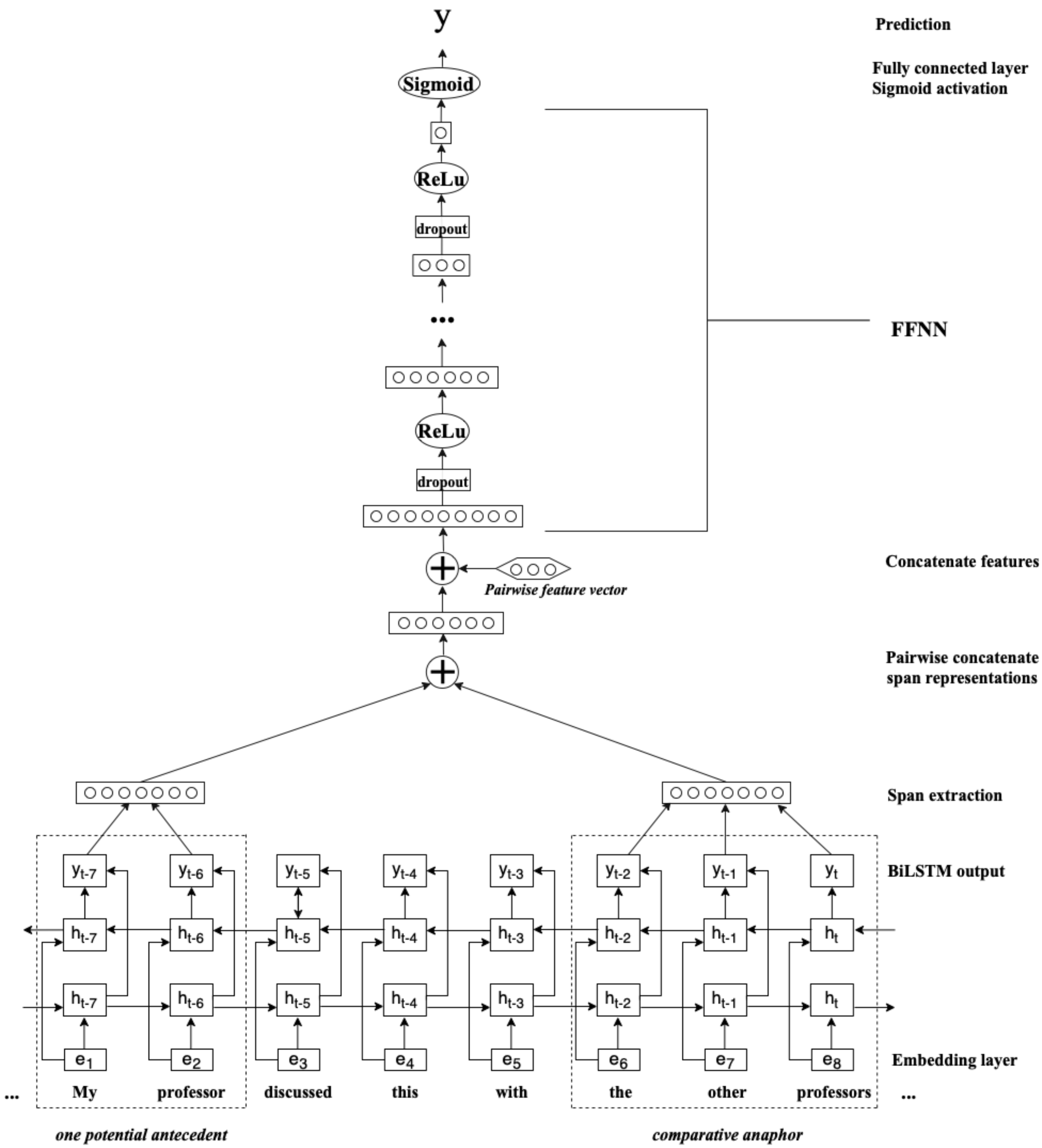
y
Prediction
Sigmoid
Fully connected layer
Sigmoid activation
ReLu
dropout
...
FFNN
ReLu
dropout
Concatenate features
Pairwise feature vector
Pairwise concatenate
span representations
Span extraction
BiLSTM output
y_{t-7} y_{t-6} y_{t-5} y_{t-4} y_{t-3} y_{t-2} y_{t-1} y_t
h_{t-7} h_{t-6} h_{t-5} h_{t-4} h_{t-3} h_{t-2} h_{t-1} h_t
e_1 e_2 e_3 e_4 e_5 e_6 e_7 e_8
Embedding layer
... My professor discussed this with the other professors ...
one potential antecedent
comparative anaphor

Figure 5.1: BiLSTM model setup

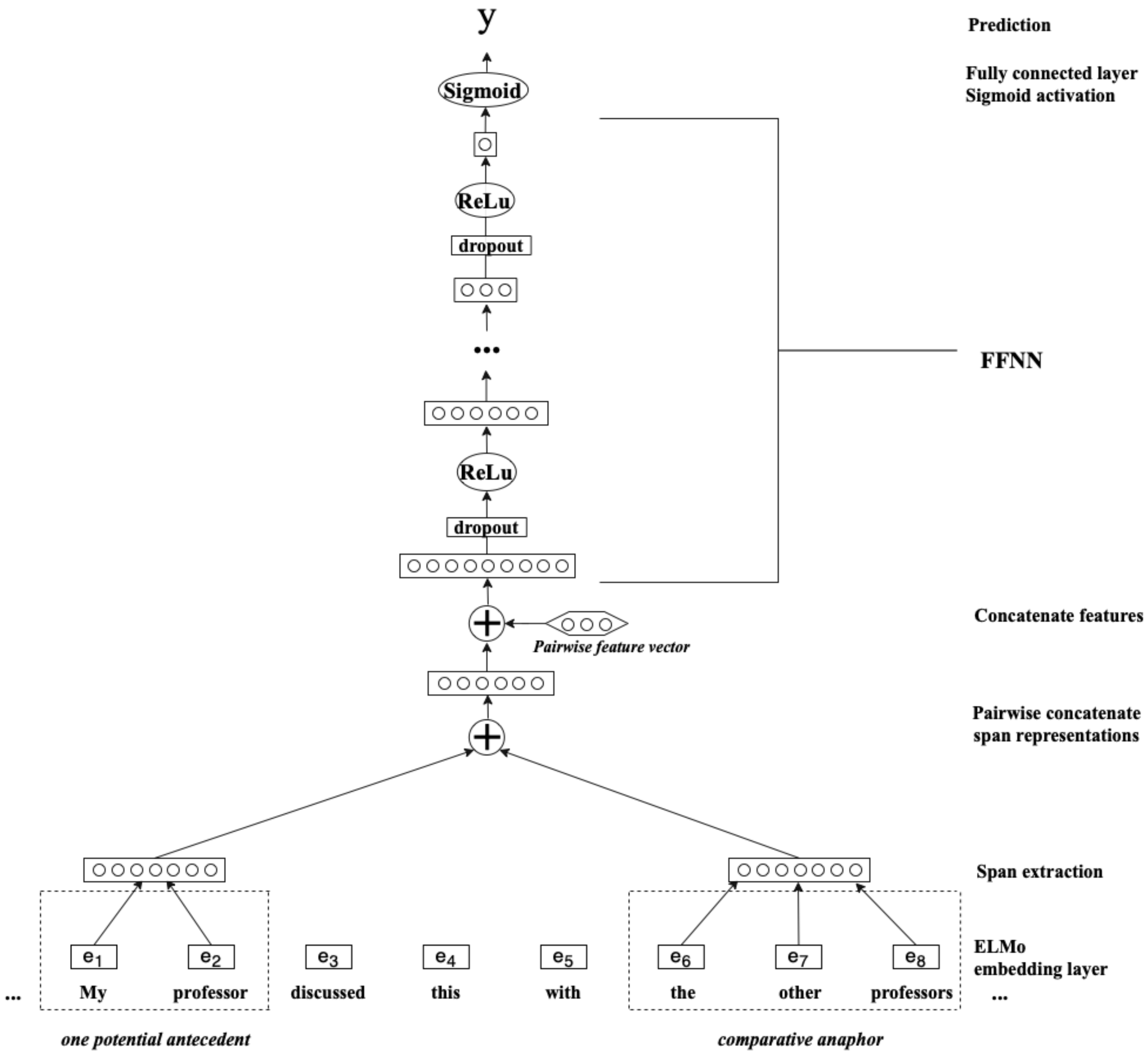

Figure 5.2: ELMo model setup

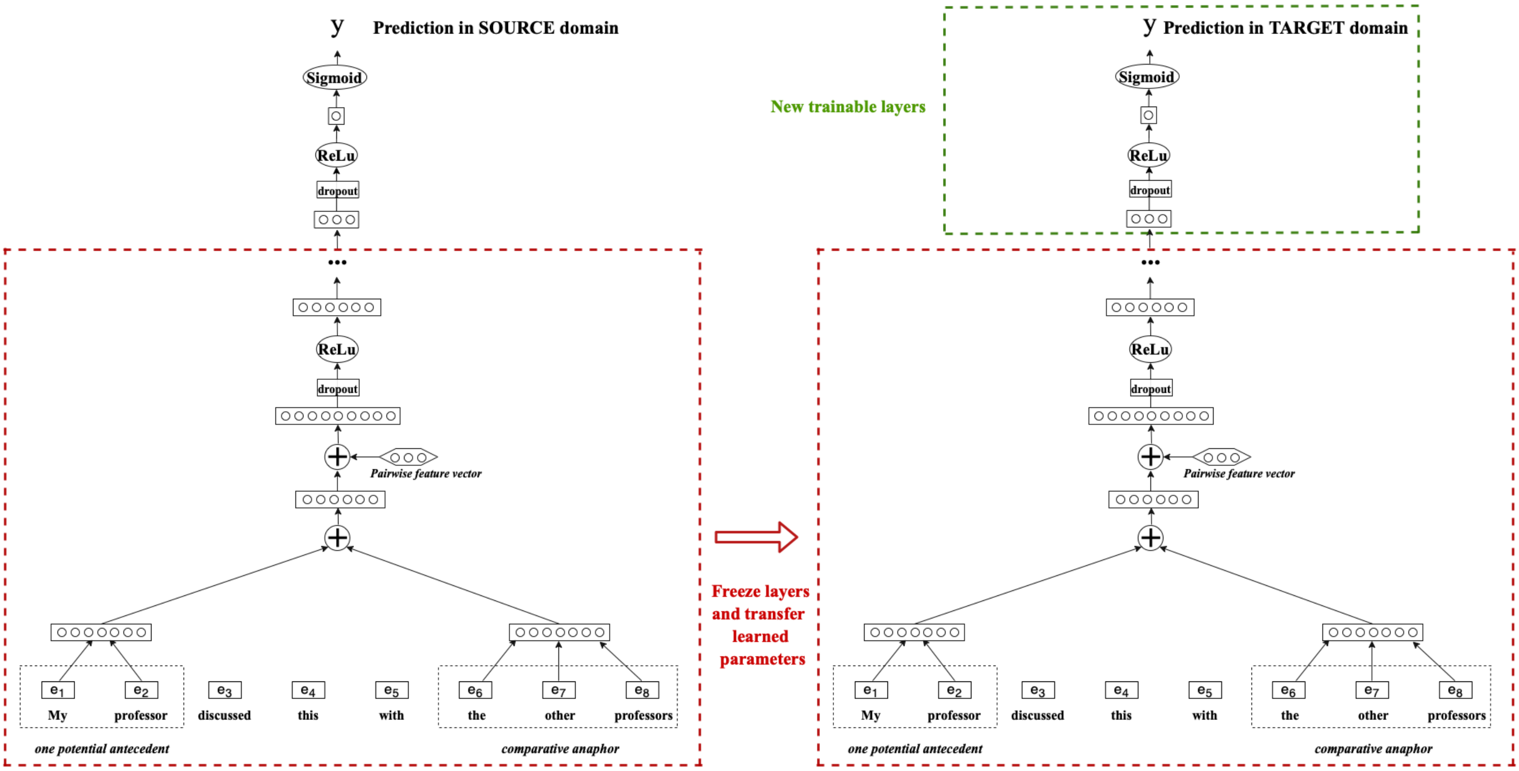

Figure 5.3: Transfer Learning - 1 (Freeze the layers) setup

Chapter 6

Experiments and Results

The BiLSTM model and the ELMo model are trained, validated, and tested respectively on WithLex and WithoutLex with ELMo embeddings, as well as the concatenation of ELMo embedding and phrase BERT phrase embedding respectively, combined with or without features. Additionally, the concatenation of ELMo and GloVe embeddings is used as input for the BiLSTM model.

6.1 Baselines

Random

The random baseline chooses a candidate from all the potential antecedents of an anaphor as the true antecedent.

Recency

The recency baseline chooses the markable that immediately precedes the anaphor as the correct antecedent.

Cosine Similarity

The cosine similarity baseline will select the candidate, whose embedding representation has the highest cosine similarity with the embedding representation of the anaphor. GloVe embeddings, ELMo embeddings, and Phrase-BERT embeddings are used respectively. For each mention span, its GloVe and ELMo embeddings representation is the summary of all word embeddings in the span along the first axis, while Phrase-BERT embeddings provide directly one embedding vector for one mention span.

In general, the success rate is always better without using max-pooling on the embeddings. Thus the success rates of all cosine similarity baselines in Table 6.1 are calculated without max-pooling.

6.2 Experiment Setups and Data Splits

All neural models are implemented in Python using PyTorch. For all experiments, Adam is used for optimization. In the feed-forward neural network, except for the last linear layer, dropout is applied to all the other linear layers (dropout=0.2). For all the models, binary cross entropy loss is used as the loss function in the training, because the main model performs binary classification. The log loss will be calculated using the formula given below:

$$L = \tfrac{1}{N}\textstyle\sum_{i=1}^{N} -(y \cdot log(p_i) + (1 - y_i) \cdot log(1 - p_i))$$

$$= \begin{cases} -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} log(1 - p_i) \text{ , if } y = 0 \\ -\dfrac{1}{N}\displaystyle\sum_{i=1}^{N} log(p_i) \text{ , if } y = 1 \end{cases} \tag{6.1}$$

Here, p_i and $(1 - p_i)$ are the candidate's probability of being the true or false antecedent. The first part of the equation will be activated, and the second part will disappear when the observation belongs to class 1, and vice versa when the observation belongs to class 0. The start learning rate of Adam is set to 1e-5.

Data Splits of Pre-training

The pre-training corpus is split in a ratio of 8:1:1 into training, validation, and test set. Data occurring in the same document are kept in the same split. Hold-out validation is used. The model is trained for 100 epochs with early stopping (patience = 8) and a batch size of 16.

Data Splits of all other Experiments

Because the comparative anaphora corpus is small, plus dividing the available data into three sets, the number of available samples is significantly reduced. The results may depend on a specific random selection of the training validation set. To avoid this, 5-fold cross-validation is applied to the training of all the models except the pre-training.

For every experiment, 5-fold cross-validation is done five times. The data is divided into five disjoint sets and each set is once a test set and once a validation set, while always keeping anaphora samples that occur in one document in the same fold. The model is trained for 80 epochs with early stopping (patience = 8) and a batch size of 16. The model with the lowest validation loss is saved as the best model.

6.3 Evaluation Measure

For every anaphor, the scores of its anaphor-candidate pairs will be ranked, and the candidate with the highest score will be selected as the prediction. Not only the gold antecedent but also its co-references are considered as true antecedent since they refer to the same real-world entity.

To evaluate the model performance, a success rate is calculated by dividing the number of all correctly resolved anaphors by the total number of all anaphors in the corpus:

$$\text{success rate} = \frac{\text{all correctly resolved anaphors}}{\text{all anaphors}}$$

For the experiments using 5-fold cross-validation, the average success rate is the average of the last iteration's success rate of every cross-validation (five cross-validations in total).

6.4 Results

The average success rates for all models are outlined in Table 6.1. Overall, for WithLex, the Phrase-BERT baseline has the highest success rate of 28.52%, while for WithoutLex, fine-tuning the pre-trained model (without any additional features or Phrase-BERT embeddings) reaches the highest success rate of 54.64%.

Baselines

The random baseline successfully predicts 13.93% of the time on the total corpus. On WithLex, it reaches a success rate of 12.36%, while on WithoutLex, it reaches a success rate of 18.71%.

The recency baseline achieves a success rate of 22.68% on the total corpus, 21.07% on WithLex, and 27.32% on WithoutLex. One reason why the recency baseline hit such nice success rates

	WithLex	**WithoutLex**	**total**
Random	12.36%	18.71%	13.93%
Recency	21.07%	27.32%	22.68%
GloVe cos. Similarity	21.40%	19.89%	21.03%
ELMo cos. Similarity	27.70%	23.78%	26.73%
Phrase-BERT cos. Similarity	**28.52**%	22.04%	26.92%
BiLSTM GloVe	10.38%	19.93%	12.73%
BiLSTM GloVe+fea.	10.56%	19.37%	12.73%
BiLSTM ELMo	11.44%	30.67%	16.18%
BiLSTM ELMo+fea.	12.31%	22.05%	14.72%
BiLSTM GloVe+ELMo	11.62%	23.12%	14.45%
BiLSTM GloVe+ELMo+fea.	11.62%	21.96%	14.17%
BiLSTM ELMo+Phrase-BERT	13.20%	34.98%	18.57%
ELMo	20.94%	**49.97**%	28.10%
ELMo+fea.	20.92%	44.11%	26.65%
ELMo+Phrase-BERT	18.48%	43.09%	24.55%
Transfer Learning (frozen)		38.13%	
Transfer Learning (fine tune)		**54.64**%	
[Klowersa 2021] BiLSTM			29.70%
[Klowersa 2021] TL			37.30%

Table 6.1: Comparison of average success rates for all models

is that 42.8% of the anaphors in the total corpus have their gold antecedent direct preceding them. (38.39% in WithLex and 55.67% in WithoutLex (see Table 4.5)).

Cosine similarity baselines also show some good results. In particular, GloVe, ELMo, and phrase BERT achieve a success rate of 21.40%, 27.7%, and 28.52% on WithLex, respectively, and they surpass the recency baseline. While on WithoutLex, all cosine similarity baselines did not surpass the recency baseline. Additionally, since the GloVe and ELMo cosine similarity baseline perform worse than the Phrase-BERT baseline on WithLex, the difference between the average cosine similarity of the embeddings of all anaphors spans with their gold antecedent (including the coreferences of gold antecedents) or their non-gold candidates on WithLex might have an important impact on the model performance on WithLex, i.e the bigger the difference, the better. (see Table 6.3).

Pre-train Model

Since the extraction of this particular pre-train corpus is rather crude, random baseline archives a success rate of 42.25% and the recency baseline shows a success rate of 52.32%. The pre-train model improves the recency baseline's success rate by 17.94%. (see Table 6.2)

	success rate
Random pre-train data	42.25%
Recency pre-train data	52.32%
Pre-train model	70.26%

Table 6.2: Pre-train results

BiLSTM Model

The BiLSTM model consistently achieves a better result on WithoutLex than on WithLex. There is no indication on why this model performs even worse than the recency baseline on Withlex. After adding features to the model, the performance on WithLex almost always improves. Interestingly, the performance on WihoutLex always deteriorates and decreases the most when using ELMo embeddings. A possible explanation for this is that the ELMo embeddings already contain all these features in the model, and the extra features just confuse the model.

The BiLSTM model provides the best success rate on both WithLex and WithoutLex corpus using the concatenation of ELMo and Phrase-BERT embeddings, with successes rate of 13.20% and 34.98%, respectively. This shows that integrating Phrase-BERT span representation improves resolving the comparative anaphora.

ELMo Model

The ELMo model performs better than the BiLSTM model, thus this model is used for the pre-training. The ELMo model performs best without adding any features or Phrase-BERT embeddings, it achieved a success rate of 20.94% on WithLex, which is better than the best result of the BiLSTM model (13.20%) but still slightly worse than the Phrase-BERT baseline (28.52%). And it achieves a success rate of 49.97% on WithoutLex. Adding feature or Phrase-BERT embeddings both makes the model perform slightly worse on both corpora. Besides, not using max-pooling provides better results than using it. This is probably because the average cosine similarity difference between anaphors and its gold antecedents (including the gold antecedent's co-references) or non-gold candidates is greater when max-pooling is not used (see Table 6.3 and 6.4).

Transfer Learning on WithLex

Transfer learning with fine-tuning provides the best success rate of 54.64%. It improves the success rate of best the ELMo model by 4.67%, while the Transfer Learning with frozen layers does not work ideally.

6.5 Error Analysis

6.5.1 Cosine Similarity Between Anaphor and Gold or Candidate Anteedents

Because the performance of the BiLSTM model on WithLex is even worse than the recency baseline, I compare the average cosine similarity of the embeddings of all anaphors spans with their gold antecedent (including the co-references of gold antecedents) or their non-gold candidates on WithLex. I also compare the difference between the average cosine similarity and whether all mention spans are max-pooled (before summing over the mention span embeddings along the first axis).

In general, the average cosine similarity difference is larger without using max-pooling than with it. Due to the very small cosine similarity difference, the model may not be able to exploit the similarity of the embeddings of two mentions.

The difference is greatest when using Phrase-BERT embedding. And the difference is then smaller when using ELMo embeddings than when using Glove embeddings.

	w/o pooling			w/ pooling		
	non gold	gold&coref.	diff	non gold	gold&coref.	diff
GloVe	0.49	0,56	0.07	0.72	0.75	0.03
ELMo	0.24	0.29	0.05	0.49	0.51	0.02
Phrase-BERT	0.42	0.53	**0.11**			

Table 6.3: Average cosine similarity between the embeddings of the anaphor and its candidates on WithLex

6.5.2 Qualitative Analysis of Predictions

The correct and incorrect predictions of different models will both be analyzed and compared. The anaphor is bold, the gold antecedent and its coreferences are in italics, and the predicted

	w/o pooling			w/ pooling		
	non gold	gold&coref.	diff	non gold	gold&coref.	diff
GloVe	0.49	0.52	0.03	0.70	0.72	0.01
ELMo	0.18	0.18	0.00	0.41	0.37	0.04
Phrase-BERT	0.44	0.52	0.08			

Table 6.4: Average cosine similarity between the embeddings of the anaphor and its candidates on WithoutLex

antecedent is underlined.

All Models' Correct Predictions

Example 1. - WithLex "If a king goes to fight against another king, first he will sit down and plan. If he has only 10, 000 men, he will try to decide if he is able to defeat the other king who has 20,000 men. If *he* thinks he can not defeat **the other king**, he will send some man to ask for peace while that king's army is still far away."

All models choose *he*, a possible explanation would be that in all three sentences, all subjects are gold antecedent or its co-references, and two of them are *he*. This may have a positive impact on the judgment of the models. Besides, out of the eleven candidates, seven are the co-references of the gold antecedent, and six are *he*. The chance of picking the correct answer is greater than 63%.

Example 2. - WithLex "*We* are able to resupply, rearm and refuel very close to the area we are working in. **No other fife navy ship** could have get in the water."

Similar to Example 1, there is only one subject in this context other than the anaphor itself, which is the correct antecedent, and there are only three candidates in total - ["the area we are working in", "we", "we"], and two of them are already the correct answer.

All Models' Incorrect Predictions

Example 3. - WithLex "Federal researchers said the lung cancer mortality rate for people under 45 years of age has begun to decline, particularly for white males. The national cancer institute also projects that the overall U.S. mortality rate from lung cancer should begin to drop in several years if cigarette smoking continues to abate. Bush met with South Korean President Roh, who indicate that *Seoul* plan to far ease trade rule to ensure that its economy become as open as **the other industrialized nation** by the mid-1990."

Almost all predictions are in the last sentence. It shows that the model learns the pattern that most of the antecedent appears in the sentence where the anaphor is located. Curiously, almost all nouns around the true antecedent were selected, even though *Seoul* is a capital of a *nation*, suggesting that all models were unable to exploit the semantic relations between two mentions. In addition, this antecedent has no co-reference, and the chance of selecting it out of 19 mentions is only about 5.3%.

Example 4. - WithLex "It prompts an IRS study that found many sellers were concealing income and treat a large amount of nondeductible travel and other personal expense as business costs.", Mr.Washburn said. The study provided criteria for single out returns of "potentially non-compliant" taxpayers who report low income and large expenses from a part-time business. The tax court recently denied business deduction by Mr. and Mrs. Peters. Rubin of Cherry Hill, N.J., who both are part-time distributors of *Amway products* in addition to their regular job as salespeople in **other fields**."

Interestingly, all ELMo models choose *part-time distributors of Amway products*, while one BiLSTM model chooses *"potentially non-compliant” taxpayers who report low income and large expenses from a part-time business*, and the rest of the BiLSTM models all chose *N.J.* We can see that the prediction of all the ELMo models is already very close to the gold antecedent. Additionally, this antecedent also has no co-reference, and there are 26 candidates for this context, the probability of randomly selecting it is only about 3.8%.

Example 5. - WithoutLex "It also says countries could temporarily raise tariffs on certain products if they experience an unusually heavy volume of imports. Instead of proposing a complete elimination of farm subsidy, as the early U.S. proposals do, the new package call for the elimination of *only the most trade-distorting one.* **Less objectionable one** would be subject only to some restraints, and others with a "relatively minor trade impact" would be allowed to continue under certain conditions."

The context of this sample is very complicated because here two anaphora intersect. *Less objectionable one* refers to *only the most trade-distorting one*, which refers to *a complete elimination of farm subsidy.* To resolute 2 anaphora at the same time might be overwhelming for all the models. Another distinctive feature of the models on the WithLex is that they more often choose very short mentions. This is likely because the first and second most frequent samples in the WithLex are pronouns and proper names, and the models seem to learn this.

WithoutLex - ELMo Model vs. BiLSTM + ELMo

Example 6. "$\underline{\text{They}}_{[BiLSTM+ELMo]}$ come from the selfish desire that makes war inside you. You want things, but you don't get them. So $\underline{\text{you}}_{[ELMo\,model]}$ kill and be jealous of **others**."

This context is not very complicated because *subject + verb + others* is a common pattern. In this case, the ELMo model resolute the antecedent *you* successfully, while the BiLSTM model using ELMo embedding as input selected *they* incorrectly, although at the very least *they* a personal pronoun. This is probably because contextualizing an already contextualized embedding (ELMo) with a BiLSTM model might produce more noise since all ELMo models consistently outperform all BiLSTM models.

WithLex - Phrase-BERT Cosine Similarity Baseline vs. BiLSTM + ELMo + Phrase-BERT

Since the Phrase-BERT baseline performs on WithLex better than the best BiLSM model, i.e. BiLSTM + ELMo + Phrase-BERT, they will be compared.

Example 7. "Republican congressional representatives, because of their belief in a minimalist state, are less willing to engage in local benefit-seeking than be democratic members of congress. If these assumptions hold, voters in races for congress face what in economic theory be called a prisoner's dilemma and have an incentive, at the margin, to lean democratic. If they put a Republican into office, not only will $\underline{\text{they}}_{[BiLSTM+ELMo+Phrase-BERT]}$ acquire less in terms of local benefit but $\underline{\textit{their selected legislator}}_{[Phrase-BERTbaseline]}$ will be relatively powerless to prevent **other legislators** from "bringing home the bacon" to their respective constituency."

In this example, the Phrase-BERT baseline selects the correct antecedent based on the similarity between the embeddings of anaphor and candidates, while the BiLSTM model selected a personal pronoun, although the correct antecedent even has the same substring *legislators* as the anaphor.

Conclusion of Error Analysis

It was observed that most of the samples that were correctly predicted by all models had often relatively few candidates, or most of the candidates were correct. It may also be beneficial if no candidate other than the correct antecedents is the subject of the model. On the contrary, the contexts of the samples where a model (or all the models) are wrongly selected are mostly relatively long with complex structures and many candidates. Besides, the cosine similarity difference of embeddings seems to have an important impact on models' performance. Besides, contextualizing an already contextualized embedding (ELMo) with a BiLSTM model might produce more noise since all ELMo models consistently outperform all BiLSTM models.

6.6 Significance

The McNemar test [McNemar 1947] is the corresponding chi-square test for paired data. The formula of the McNemar test shows that it is only based on the pairs where there is a disagreement between two models. The null hypothesis is that is the two models perform differently. If the p-value (probability value) of a pair of models is smaller than the threshold of 0.05, the hypothesis is true, i.e. it is considered significant, implying that these models have significantly different proportions of incorrect predictions.

The McNemar test results in Table 6.5 compare prediction errors between the recency baseline and all models on WithoutLex, between the Phrase-BERT baseline and all models on WithLex, between the ELMo model and the transfer learning, as well as between all models using and not using features on the whole corpus. The two bold results are the smallest p-values on two separate corpora respectively.

For correctly resolving any anaphora on WithoutLex, all ELMo models perform significantly better than the recency baselines. However, all BiLSTM models either perform significantly worse than the recency baseline or do not perform significantly differently from the recency baseline. On WithLex, the Phrase-BERT cosine similarity baseline always performs significantly better than all models.

On WithoutLex, the ELMo model performs significantly better without features than with features, and the other models did not perform significantly differently without and with features. While on WithLex the BiLSTM + GloVe with features performs significantly better than without features, and there is no significant difference in the performance of all other models with and without features. This proves that almost all models do not make good use of features to improve performance. Besides, the transfer learning with fine-tuning outperforms the ELMo model, while the transfer Learning with frozen layers performs worse than the ELMo model. Both results are significant.

model 1	**model 2**	**p value**	**p<0.05**
WithoutLex			
Recency	ELMo*	5.66E-06	True
Recency	ELMo+fea.*	**9.01E-04**	True
Recency	ELMo+Phrase-BERT*	9.14E-01	True
Recency*	BiLSTM GloVe	5.81E-02	False
Recency*	BiLSTM GloVe+fea.	8.64E-03	True
Recency*	BiLSTM ELMo	9.05E-01	False
Recency	BiLSTM ELMo+fea.*	9.04E-01	False
Recency*	BiLSTM GloVe+ELMo	2.28E-01	False
Recency*	BiLSTM GloVe+ELMo+fea.	3.63E-02	True
Recency	BiLSTM ELMo+Phrase-BERT*	1.65E-01	False
ELMo*	ELMo+fea.	4.33E-02	True
BiLSTM GloVe*	BiLSTM GloVe+fea.	1.80E-01	False
BiLSTM ELMo	BiLSTM ELMo+fea*	5.97E-01	False
BiLSTM GloVe+ELMo*	BiLSTM GloVe+ELMo+fea	3.07E-01	False
ELMo*	TF$_{frozen}$*	3.86E-01	True
ELMo	TF$_{finetune}$*	4.22E-05	True
WithLex			
Phrase-BERT*	ELMo	2.26E-03	True
Phrase-BERT*	ELMo+fea.	3.90E-03	True
Phrase-BERT*	ELMo+Phrase-BERT	1.33E-04	True
Phrase-BERT*	BiLSTM GloVe	**1.81E-14**	True
Phrase-BERT*	BiLSTM GloVe+fea.	2.98E-08	True
Phrase-BERT*	BiLSTM ELMo	6.85E-11	True
Phrase-BERT*	BiLSTM ELMo+fea.	1.26E-09	True
Phrase-BERT*	BiLSTM GloVe+ELMo	1.83E-12	True
Phrase-BERT*	BiLSTM GloVe+ELMo+fea.	1.36E-12	True
Phrase-BERT*	BiLSTM ELMo+Phrase-BERT	3.77E-10	True
ELMo	ELMo+fea*	8.92E-01	False
BiLSTM GloVe	BiLSTM GloVe+fea.*	1.34E-02	True
BiLSTM ELMo	BiLSTM ELMo+fea.*	7.95E-01	False
BiLSTM GloVe+ELMo*	BiLSTM GloVe+ELMo+fea.*	1	False

Table 6.5: McNemar test results comparing different model pairs (The better performing model in a pair, i.e. the one with fewer errors, will be marked with an asterisk.)

Chapter 7

Conclusion

This work applied different neural models to resolve comparative anaphora with or without lexical heads. One model includes BiLSTM architecture and another mainly utilizes ELMo architecture. Both models make use of contextual and external knowledge.

Because the comparative anaphora corpus is relatively small, I created a pronominal anaphora dataset for pre-training, since pronominal anaphora are similar to comparative anaphora without lexical heads. The ELMo model is first pre-trained on the pronominal anaphora resolution task. Then, two different types of transfer learning were employed to the pre-trained model on WithoutLex, i.e. either the pre-trained model will be fine-tuned or its layers will be frozen.

Some of the initial hypotheses were confirmed and some were not:

1. There are significant differences between comparative anaphora with and without lexical heads since the models perform very differently on WithLex and Withoutlex. Besides, the statistical analysis show differences.

2. Resolving the antecedent of anaphor without lexical heads is extremely context-dependent since the ELMo models perform better than the BiLSTM models on WithoutLex.

3. Including extra features only slightly improves the BiLSTM model's performance on WithLex.

4. Pre-training on pronominal anaphora data can improve the model performance for the anaphora resolution task on WithoutLex.

5. Additionally concatenating Phrase-BERT embedding only improve the BiLSTM model's performance.

A lot of possible future work can be done to further improve the model performance — My pre-train corpus could be further purified, for example by removing reflexive pronouns. Moreover, features can be used purposefully, such as not using semantic features on WithoutLex or developing some other different features. Besides, creating a dataset with much more high-quality comparative anaphora data could improve the generalization ability of the models. To explore whether the model is also valid for this task in other languages, the model can be trained on other comparative anaphora datasets in different languages. Fine-tuning the pre-trained BERT model as [Klowersa 2021] can be also considered since it performs better than her BiLSTM model.

In conclusion, the two main contributions of this thesis are the proof of the difference between anaphora with and without lexical heads, as well as the similarity between pronominal anaphora and anaphora without heads. Besides, Transfer Learning is also proven in this study to be an effective method for resolving comparative anaphora.

List of Figures

2.1 Recurrent Neural Network workflow 9

2.2 A BiLSTM cell 11

2.3 BiLSTM model workflow 12

2.4 ELMo uses a multi-layer biLM 14

5.1 BiLSTM model setup 32

5.2 ELMo model setup 33

5.3 Transfer Learning - 1 (Freeze the layers) setup 34

List of Tables

2.1 Activation functions referenced in this work with their derivative and range . . . 8

4.1 Number of anaphors in the total comparative anaphora corpus 22

4.2 Genre of the anaphors in the total comparative anaphora corpus 22

4.3 Distribution of the split corpus 22

4.4 Quantitative statistics of data 23

4.5 Ratio of gold antecedents in same and last n sentences preceding the anaphor . . 23

4.6 Frequency of the antecedents being a subject 23

4.7 Distribution of linguistic types of the gold and candidate antecedents 24

4.8 Quantitative statistics of pre-train data and WithoutLex 25

4.9 Ratio of gold antecedents in same and last n sentences preceding the anaphor in the pre-train data 25

5.1 Features used for training 27

6.1 Comparison of average success rates for all models 38

6.2 Pre-train results 39

6.3 Average cosine similarity between the embeddings of the anaphor and its candidates on WithLex 40

6.4 Average cosine similarity between the embeddings of the anaphor and its candidates on WithoutLex 41

6.5 McNemar test results comparing different model pairs (The better performing model in a pair, i.e. the one with fewer errors, will be marked with an asterisk.) 45

bibliography

Bibliography

Chen, Bin, Xiaofeng Yang, Jian Su, and Chew Lim Tan (2008). "Other-anaphora resolution in biomedical texts with automatically mined patterns." In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 121–128.

Chen, Chen and Vincent Ng (2016). "Chinese zero pronoun resolution with deep neural networks." In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 778–788.

Clark, Herbert H (1975). "Bridging." In: *Theoretical issues in natural language processing.*

Cruse, DA (1980). "John A. Hawkins, Definiteness and indefiniteness: a study in reference and grammaticality prediction. London: Croom Helm, 1978. Pp. 316." In: *Journal of Linguistics* 16.2, pp. 308–316.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.* arXiv: 1810.04805 [cs.CL].

Elman, Jeffrey L. (1990). "Finding structure in time." In: *Cognitive Science* 14.2, pp. 179–211. ISSN: 0364-0213. DOI: https://doi.org/10.1016/0364-0213(90)90002-E. URL: https://www.sciencedirect.com/science/article/pii/036402139090002E.

Fellbaum, Christiane D. (2000). "WordNet : an electronic lexical database." In: *Language* 76, p. 706.

Grosz, Barbara J, Aravind K Joshi, and Scott Weinstein (1995). "Centering: A framework for modelling the local coherence of discourse." In.

Halliday, M A. K and Ruqaiya Hasan (1976). *Cohesion in English.* London: Longman. ISBN: 1897618034 9781897618035.

Hermann, Karl Moritz, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom (2015). “Teaching machines to read and comprehend.” In: *Advances in neural information processing systems* 28, pp. 1693–1701.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). “Long short-term memory.” In: *Neural computation* 9.8, pp. 1735–1780.

Honnibal, Matthew and Ines Montani (2017). “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing.” To appear.

Hou, Yufang, Katja Markert, and Michael Strube (2014). “A rule-based system for unrestricted bridging resolution: Recognizing bridging anaphora and finding links to antecedents.” In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2082–2093.

Klowersa, Miriam (2021). “Neural Methods for Comparative Anaphora Resolution.” In.

Krishna, Kalpesh, John Wieting, and Mohit Iyyer (2020). *Reformulating Unsupervised Style Transfer as Paraphrase Generation.* arXiv: 2010.05700 [cs.CL].

Liu, Ting, Yiming Cui, Qingyu Yin, Weinan Zhang, Shijin Wang, and Guoping Hu (2016). “Generating and exploiting large-scale pseudo training data for zero pronoun resolution.” In: *arXiv preprint arXiv:1606.01603*.

Loper, Edward and Steven Bird (2002). “NLTK: The Natural Language Toolkit.” In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP ’02. Philadelphia, Pennsylvania, pp. 63–70. DOI: 10.3115/1118108.1118117. URL: https://doi.org/10.3115/1118108.1118117.

Markert, Katja, Yufang Hou, and Michael Strube (July 2012). “Collective Classification for Fine-grained Information Status.” In: pp. 795–804. URL: https://aclanthology.org/P12-1084.

McNemar, Quinn (June 1947). “Note on the sampling error of the difference between correlated proportions or percentages.” In: *Psychometrika* 12.2, pp. 153–157. DOI: 10.1007/bf02295996. URL: https://doi.org/10.1007/bf02295996.

Modjeska, Natalia Nygren (2003). “Resolving other-anaphora.” PhD thesis. University of Edinburgh.

Modjeska, Natalia Nygren, Katja Markert, and Malvina Nissim (2003). “Using the web in machine learning for other-anaphora resolution.” In: *Proceedings of the 2003 conference on Empirical methods in natural language processing*, pp. 176–183.

Ng, Vincent and Claire Cardie (2002). “Improving machine learning approaches to coreference resolution.” In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 104–111.

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). “GloVe: Global Vectors for Word Representation.” In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543. URL: http://www.aclweb.org/anthology/D14-1162.

Peters, Matthew E., Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer (2018). *Deep contextualized word representations*. cite arxiv:1802.05365Comm NAACL 2018. Originally posted to openreview 27 Oct 2017. v2 updated for NAACL camera ready. URL: http://arxiv.org/abs/1802.05365.

Reimers, Nils and Iryna Gurevych (2019). *Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks*. arXiv: 1908.10084 [cs.CL].

Riester, Arndt and Stefan Baumann (Jan. 2017). “The RefLex Scheme – Annotation Guidelines.” In: DOI: 10.18419/opus-9011.

Rösiger, Ina (2018). “Rule-and learning-based methods for bridging resolution in the ARRAU corpus.” In: *Proceedings of the First Workshop on Computational Models of Reference, Anaphora and Coreference*, pp. 23–33.

Rösiger, Ina, Arndt Riester, and Jonas Kuhn (2018). “Bridging resolution: Task definition, corpus resources and rule-based experiments.” In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3516–3528.

Ruslan, Mitkov and Wolverhampton Sb (2000). “Anaphora Resolution: The State Of The Art.” In.

Schuster, Mike and Kuldip Paliwal (Dec. 1997). “Bidirectional recurrent neural networks.” In: *Signal Processing, IEEE Transactions on* 45, pp. 2673–2681. DOI: 10.1109/78.650093.

Soon, Wee Meng, Hwee Tou Ng, and Daniel Chung Yong Lim (2001). “A machine learning approach to coreference resolution of noun phrases.” In: *Computational linguistics* 27.4, pp. 521–544.

Uryupina, Olga, Ron Artstein, Antonella Bristot, Federica Cavicchio, Francesca Delogu, Kepa J Rodriguez, and Massimo Poesio (2020). “Annotating a broad range of anaphoric phenomena, in a variety of genres: the ARRAU corpus." In: *Natural Language Engineering* 26.1, pp. 95–128.

Wang, Shufan, Laure Thompson, and Mohit Iyyer (2021). *Phrase-BERT: Improved Phrase Embeddings from BERT with an Application to Corpus Exploration.* arXiv: 2109.06304 [cs.CL].

Werbos, P.J. (1990). "Backpropagation through time: what it does and how to do it." In: *Proceedings of the IEEE* 78.10, pp. 1550–1560. DOI: 10.1109/5.58337.

Zhang, Hongming, Yan Song, and Yangqiu Song (June 2019). "Incorporating Context and External Knowledge for Pronoun Coreference Resolution." In: pp. 872–881. DOI: 10.18653/v1/N19-1093. URL: https://aclanthology.org/N19-1093.

Zimmermann, Victor. (2019). "Neural resolution of comparative anaphora(unpublished undergraduate thesis)." In: *University of Heidelberg.*