Evolutionary Discovery of Natural-Language Coreference Chains for Social Media Analysis

John Atkinson (john.atkinson@uai.cl)
Universidad Adolfo Ibañez

Alex Escudero
Beacon42 Ltd

Research Article

Keywords: Evolutionary, Discovery, Natural-Language, Coreference Chains, Social Media Analysis, linguistic knowledge

Posted Date: June 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-562748/v1

Evolutionary Discovery of Natural-Language Coreference Chains for Social Media Analysis

John Atkinson
Universidad Adolfo Ibañez, Santiago, Chile. john.atkinson@uai.cl
Alex Escudero
Beacon42 Ltd, Santiago, Chile. alex.escudero@beacon42.com
Correspondence to john.atkinson@uai.cl and John Atkinson

June 16, 2021

Abstract

Communicating messages on social media usually conveys much implicit linguistic knowledge, which makes it difficult to process texts for further analysis. One of the major problems, the linguistic co-referencing task, has mostly been addressed for formal and full-sized text in which a relatively clear discourse structure can be discovered, using Natural-Language Processing techniques. However, texts in social media are short, informal and lack a lot of underlying linguistic information to make decisions so traditional methods can not be applied. Furthermore, this may significantly impact the performance of several tasks on social media applications such as opinion mining, network analysis, sentiment analysis, text categorization. In order to deal with these issues, this research address the task of linguistic co-referencing using an evolutionary computation approach. It combines discourse coreference analysis techniques, domain-based heuristics (i.e., syntactic, semantic and world knowledge), graph representation methods, and evolutionary computation algorithms to resolving implicit co-referencing within informal opinion texts. Experiments were conducted to assess the ability of the model to find implicit referents on informal messages, showing the promise of our approach when compared to related methods.

1 Introduction

There is a huge amount of implicit linguistic knowledge on natural language documents that needs to be addressed in order to deal with several text analytics tasks so as to discover intents, discourse entities, referents, to conduct robust *Natural-Language Processing* (NLP). In the context of computational discourse analysis, linguistic models and techniques have been used to successfully understanding the intent and meaning of a phrase or discourse via intentions,

goals, and underlying rhetorical relations within a natural-language text. One of the crucial tasks is to resolve references to implicit entities via pronouns, and lexical chains by detecting expressions that refer to the same entity within a text, aka, co-reference resolution. This effort is heavily used in tasks such as question-answering, information extraction, document summarization, sentiment analysis. Co-reference or anaphora resolution techniques [3] have been developed firstly in a very robust linguistic knowledge context. However, representing and processing this information are very complex tasks hence research evolved toward *Machine Learning* (ML) techniques that minimize human input and maximize model robustness by taking advantage of state-of-the-art methods so as to enable applications using informal text data such as text categorization, lexical tagging and named-entity recognition.

There are some efficient lexical co-referencing methods for discourse analysis which are usually successful on formal and long (full-sized) natural-language texts [12]. However, no significant evidence has been found to effectively perform this task for informal and short text messages lacking much linguistic knowledge. This is usually the case for texts found on SMS, social media, product reviews, tweets. In particular, on social media, unprecedented opportunities for people have been created to publicly voice their opinions expressing underlying sentiments and emotions on politicians, those sentiment is usually referred to as *sentiment analysis*.

For example, in order to analyse messages on tweets, one must first be capable of getting relevant opinions in order to identify information such as the opinion's polarity on a tareget topic. A key problem is that search engines on opinions assume that target objects are explicitly expressed in a text as a set of keywords. However, social media messages convey a lot of implicit knowledge on the entities being mentioned. This may cause that opinions containing different entities (but conceptually equivalent) are seen as different ones between each other due to the implicit ways different entities/objects can be referenced.

For example, assume different messages containing the entities such: `proposal`, `the document` (e.g., a lexical co-referent for `proposal`) and `it` (e.g., a pronoun anaphora for `the document` and `proposal`), so they are all referring to the same concept (aka., synonyms). However, when retrieving opinions on `proposal`, the other message will be missed by a search engine. As a consequence, since the task fails to understand the synonym relationship, the quality of a further sentiment analysis task will drop. Despite the impact of this task, current sentiment analysis techniques do not address this issue, resulting in reduced *recall* of the sentiment classifiers. Accordingly, in this paper a new evolutionary computation model for discovering language co-references in short and informal texts is proposed for sentiment analysis applications. It combines evolutionary optimization and graph-based representations in order to detect co-reference or anaphora patterns which are in turn assessed by a *Genetic Algorithm* (GA). It uses linguistic context and a weights system that are assessed by a domain-dependent heuristic in order to discover co-reference chains.

This paper is organized as follows: section 2 discusses the main foundations and related work on linguistic co-referencing and anaphora resolution methods

in NLP, section 3 describes our evolutionary computation method for anaphora resolution on social media using graph representation and linguistic heuristis for dealing with informal and short texts, section 4 discusses a series of experiments conducted for our linguistically-motivated evolutionary computation method for anaphora resolution from micro-blogging systems, and finally, 5 highlights the main conclusions and outcomes of this research.

2 Related Work

There is some significant research on lexical co-reference and anaphora resolution approaches for formal and long-sized texts, usually heavily depending on a specific target natural language. These methods can usually model a text's discourse structure in order to further look for co-reference or anaphoric links. However, there is no such a fair discourse structure within short texts found in micro-blogging systems (i.e., twitter), e-commerce reviews, so state-of-the-art techniques cannot usually be applied. This is a key issue when dealing with tasks such as *Sentiment Analysis* (SA) in which a sentiment must be detected for an entity (i.e., product, person, organization) that in most cases can be implicit in the opinion/tweet/message. Hence traditional SA methods are not effective as their text representation models usually assume a typical bag-of-words representation in which key entities must be explicitly expressed. Furthermore, the length of the texts (i.e., usually no more than 2 sentences on twitter) restrict the task from performing deep linguistic processing. These methods can usually model a text's discourse structure in order to further look for co-reference or anaphoric links. However, there is no such a fair discourse structure within short texts found in micro-blogging systems (i.e., twitter), e-commerce reviews, so state-of-the-art techniques cannot usually be applied. This is a key issue when dealing with tasks such as *Sentiment Analysis* (SA) in which a sentiment must be detected for an entity (i.e., product, person, organization) that in most cases can be implicit in the opinion/tweet/message. Hence traditional SA methods are not effective as their text representation models usually assume a typical bag-of-words representation in which key entities must be explicitly expressed. Furthermore, the length of the texts (i.e., usually no more than 2 sentences on twitter) restrict the task from performing deep linguistic processing. These methods can usually model a text's discourse structure in order to further look for co-reference or anaphoric links. However, there is no such a fair discourse structure within short texts found in micro-blogging systems (i.e., twitter), e-commerce reviews, so state-of-the-art techniques cannot usually be applied. This is a key issue when dealing with tasks such as *Sentiment Analysis* (SA) in which a sentiment must be detected for an entity (i.e., product, person, organization) that in most cases can be implicit in the opinion/tweet/message. Hence traditional SA methods are not effective as their text representation models usually assume a typical bag-of-words representation in which key entities must be explicitly expressed. Furthermore, the length of the texts (i.e., usually no more than 2 sentences on twitter) restrict the task from performing deep lin-

guistic processing. These methods can usually model a text's discourse structure in order to further look for co-reference or anaphoric links. However, there is no such a fair discourse structure within short texts found in micro-blogging systems (i.e., twitter), e-commerce reviews, so state-of-the-art techniques cannot usually be applied. This is a key issue when dealing with tasks such as *Sentiment Analysis* (SA) in which a sentiment must be detected for an entity (i.e., product, person, organization) that in most cases can be implicit in the opinion/tweet/message. Hence traditional SA methods are not effective as their text representation models usually assume a typical bag-of-words representation in which key entities must be explicitly expressed. Furthermore, the length of the texts (i.e., usually no more than 2 sentences on twitter) restrict the task from performing deep linguistic processing.

Yet, well-known co-reference resolution methods for English were observed to fail when attempting to resolve ambiguous gender phrases as the language uses the same article/determiner for both genders. For example, one may assume that for the sentence `'the secretary'`, the subject should be a female genre so it should be referred to as `she/her`. However, when one turns into languages different from English (i.e., Spanish), the rule does not hold as there is a different determiner for each gender (i.e., `El` for his and `La` for her). On the contrary, English pronouns tell us a lot about an entity being referred (i.e., `it, their, her, his`) whereas for Spanish, a single pronoun (i.e., `su`) has an inherent ambiguity that can refer to various candidate entities (i.e. the thing, her thing, his thing, their thing) hence its increasing complexity of the co-referencing task.

Early approaches to anaphora resolution used syntactic and semantic heuristics assessing entities and referents in every sentence of a full text, making sense of the relationship between them so that these can be evaluated as candidate entities for co-reference chains. For instance, number and gender can be compared directly between words, so it is fairly simple differentiating one referent from another. Some further research incorporated heuristics for reflexive, reciprocal and pleonastic anaphora. These domain-specific heuristics can be refined in certain contexts of dialogue models [5, 6, 11]. Thus, modern approaches used rule-based strategies for a wide variety of anaphora such as pronouns, reflexives and deictic anaphora in multi-person dialogues represented as anaphoric chains (i.e., chains of referent candidates) with knowledge-poor constraints and heuristics which are then fine-tuned using a heuristic optimization method based on evolutionary computation (i.e., Genetic Algorithms).

These rules work on a naive character model of an entire dialogue which is represented as a graph where nouns and pronouns are extracted, and links between candidate referents and entities are looked for by exploring several path-based properties [6]. Thus, the graph represents a set of candidate anaphora-antecedent relationships where the group of candidate referents of an anaphora represented by a 'pronoun node' consists of all possible distinct 'noun nodes' that can be reached using paths satisfying some properties (e.g., paths above the length of two nodes represent anaphoric chains in the dialogue). An antecedent space of an anaphora consists of all nouns and pronouns whose corresponding nodes are reachable from the 'pronoun node' by traversing a single edge and this

antecedent space is processed in a way that nodes in the chain are ranked to determine which of these is the best candidate for the anaphora in question. An advantage of the approach is that finding the best lexical co-reference/anaphora chains can be seen as a single optimization problem in which a large search space of candidate links must effectively be explored by applying some domain-dependent operations and assessing the solutions by using linguistic heuristics. Experiments show the methods achieved an accuracy of 65% on producing correct anaphora links on dialogue samples.

Meta-heuristic optimization techniques using GAs have also been applied to perform pronoun resolution on formal language sentences extracted from the Treebank corpus (https://catalog.ldc.upenn.edu/LDC99T42). The approach combines syntactic and semantic analysis techniques to determine the antecedent of pronouns by casting votes on those techniques, which are assigned a weight so that the candidate antecedent which receives the most votes is selected. GAs are used to find the optimal weight assignment for each linguistic technique when assessing candidate solutions, with the fitness evaluation being the proportion of anaphora links correctly resolved. As a result, while the method resolved the task with an accuracy of 69%. it was achieved by using full-sized and formal natural-language texts.

Traditional methods such as MARS [21] have also been adapted by using evolutionary computation techniques. These first apply a filter so that no NP (Noun Phrase) may be considered as a candidate for the antecedent of a pronoun if it does not agree with the pronoun in terms of number and gender. A set of boosting and impeding indicators (i.e., heuristics) is then applied to each candidate NP depending on a pronoun's candidates. Fitness evaluation is conducted by summing up the number of anaphora correctly resolved by the system whenever the original indicator outcome values are replaced by those generated by the GA. The system was originally applied to resolve anaphora links (i.e., chains) on a few technical manuals, achieving an accuracy of 97% when compared to human performance. Nevertheless, indicators may contradict each other so the results become very misleading. Furthermore, the sizse of the sampled texts are not significant so as to provide a robust anaphora resolution task.

Supervised ML techniques such as decision trees were also explored to pairwise classification of pairs of entity mentions by using graph inference methods [19]. Once they are classified, single-link clustering is performed to produce final co-referential chains [22]. This graph-based approach for anaphora resolution [8] has shown promise to mapping a set of references to entities into a minimal collection of individual entities, where each entity mention in a text is represented as a vertex of a graph, and edges are added to the graph for every pair of vertices representing mentions which can potentially be the same entity. A set of constraints between two mentions is used to compute a weight value in each edge which is used to assess entity-pronoun pairs. Thus, for each set of mentions to resolve, a vertex is added to the graph and attributes (i.e., genre, number) are connected to each mention set, so heuristics and constraints can be applied to assess the solution and then weights are assigned to the graph based on the defined constraints [13, 24].

The task is then conducted by using probabilistic and deterministic learning algorithms such as *Relaxation Labeling* on different partitions of the graph [8]. A disadvantage of the method is that in order to achieve a good performance, the right combination of constraint weights must be found, which is a very time and expertise demanding task. Experiments using the ACE corpus (https://www.ldc.upenn.edu/collaborations/past-projects/ace) which is composed of broadcast news, newswire and newspaper content, achieved an accuracy of 69.5%. Recently, *Memory Based Learning* (MBL) has been used to infer co-reference chains on micro-blogging messaging platforms such as *twitter*. The lack of linguistic information on tweets has been addressed by taking advantage of the twitter's hierarchy tree in its discussion threads [3]. The MBL-based classifier is trained on the ANCORA corpus (http://clic.ub.edu/corpus/es/ancora) by using pairs of entities and referents, and takes into account the repetition of entities lying in the classical forum thread hierarchies, and also uses underlying special-purposes symbols such as hash-tags in order to provide the method with additional linguistic information so as to find co-reference chains. Using cross-validation methods and formal and informal short texts, the approach achieved an F-score = 0.74. Furthermore, in spite of the lack of large annotated training corpus, unsupervised learning approaches have also been used for the target task by combining clustering techniques and decision trees classification on formal texts (i.e., news from MUC-6 and MUC-7 datasets containing 54.888 words and 58.594 words, respectively [18]. Results are promising in terms of the achieved precision (78.0) but it still has low recall (64.2), mainly due to the way the linguistic features are coded manually. Further improvements of these methods by using the Ontonotes (https://catalog.ldc.upenn.edu/LDC2013T19) training corpus on twitter conversations and the MUC-7 dataset [2] have showwn no significant increase in performance (precision=74, recall=62) due to the nature of the informal messages on social media and the sub-language usually contained in twitter which is note the same as that in Ontonote [15, 10].

New models based on word embeddings and deep neural network approaches seem to perform well with complex tasks such as classification and sentiment analysis. However, regardless of how these models are trained, they do not appear to learn any form of general semantic comprehension. In other words, the models are not general language models, instead performing well on a broad range of tasks, namely those that they were trained to perform well on [9]. Thus we can think of these methods as 'next-word' predictors. Because of huge datasets required for training, running these NLP models involves using a lot of GPU resources and hours (or even days) of training, which are obviously not available for our co-referencing task [4, 7, 14, 17].

3 A GA-based Graph Approach for Co-reference Resolution on Social Media

Messaging on social media has some significant constraints that prevent it from using traditional NLP approaches to detect knowledge on implicit entity mentions such as lexical co-reference chains. Some of these limitations include:

1. Tweets are usually restricted to 280 characters, however, the average length does not exceed 33 characters. This means one does not have too much room to express full ideas and properly mention entities.
2. There are small training datasets available for very domain-specific tweets so DL methods or even pretrained and transfer learning methods cannot be applied.
3. Messages are intended as an informal communication channel so people use many idioms, slangs, abbreviations and special-purpose symbols so that *tweets* can fit the short provided space.

In addition, whenever a discussion thread is started (i.e., users replying to other users), hash-tag symbols can be used to directly point at certain discussion entities such as topics or users (i.e., @*topic* or #*user*). While this not a natural way of referencing in natural language and can become very ambiguous, it may provide us with a set of candidate entities when solving co-references and anaphora. In order to address these issues, in this research a new co-reference resolution method is proposed to deal with implicit mentions in social media, by combining evolutionary computation techniques and graph-based representation of messages, so as to feed further tasks such as sentiment analysis, text categorization. The approach aims at recognizing referents on standard natural language but using 'assistance' provided by the twitter thread hierarchy and *twitter*'s special-purpose symbols and mentions.

The approach adapts a graph-based approach for representing candidate co-reference chains and entity links extracted from an annotated corpus. Candidate entities are built from specific *twitter*'s hierarchy of threads on a given topic. A GA is then applied to find the best chains (sequences) of referents and entities by using specific-purpose linguistic criteria.

Our model's architecture can be seen in figure 1 in which a tweets dataset is collected from a given input topic based on local news, which are manually annotated for co-reference purposes (aka., annotated corpus). It then performs text preprocessing tasks in order to extract lexical and syntactical information to represent messages in the form of (candidate) graph relations. A GA then iteratively searches for and optimizes the best graphs representing co-referencing chains based on the reference annotated corpus and defined linguistic heuristics.

3.1 Annotated Corpus Collection

In order to collect a working corpus, the twitter API is used to download tweet threads on a given topic based on local news. These are arranged as a tree hier-

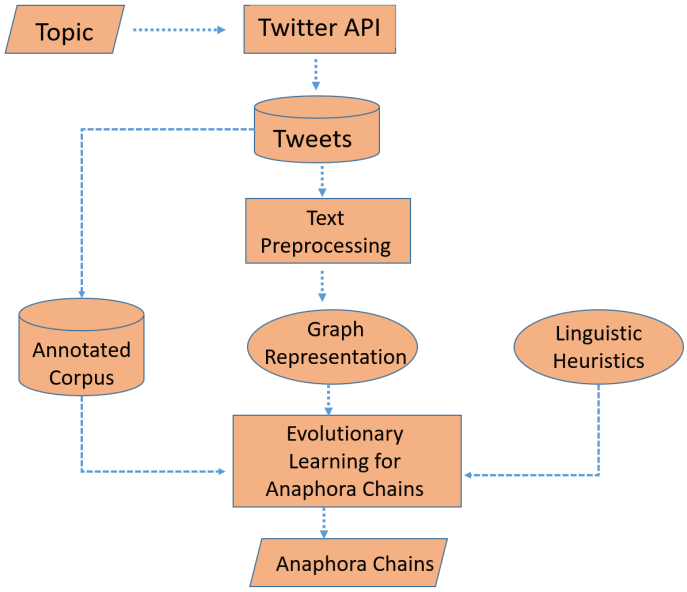


Figure 1: Arquitecture for our Twitter-based Evolutionary Model for Anaphora Resolution

archy which represents the conversation thread between a tweet (i.e., opinion) and its replies. The corpus is then manually annotated for further use in the optimization step.

3.2 Text Preprocessing

Twitter's threads are cleaned up so as to remove strange characters, leaving only natural language and twitter slangs and symbols. The tweets corpus was then annotated with every entity and referent mentioned in the texts. A lexicon of custom terms and expressions was also created for further tagging tasks, so they can be recognized as a part of the language (i.e., pronouns and nouns). POS tagging is then applied to the normalized corpus in order to identify the role of each word in a sentence. Mentions are treated as nouns as they refer mostly to entities, whereas a hash-tag was treated as a single token having some probability of being either a noun or an adjective.

In order to create a list of candidate entities, a *Named-Entity Recognition* (NER) task is applied to corpus to recognize useful multi-word entities such as organizations, people, locations, from lexical information provided by the POS tagger. Relationships between entities/words in every sentence are then analyzed by using a *dependency parser* in order to extract a grammatical structure to be represented as a graph. The dependency parser analyzes dependency relationship within a sentence between head words and words which modify those

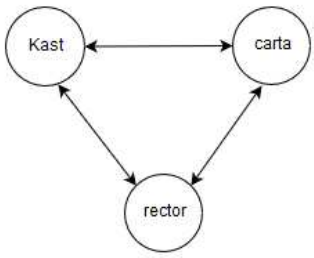
heads. These relationships will build up the connections between nouns and pronouns in the graph. This aims at mapping a set of references to entities into a minimal collection of individual entities, by grouping candidate co-reference chains.

3.3 Graph Representation

Once twitter entities and relationships have been identified, an adjacency table (figure 2(a)) is generated for every pair of sentences and then the corresponding graph representation (figure 2(b)) is produced from dependency relations.

From	To	Token	POS	F1	F2
S001	S001	Kast	PROPN	Gender=male	Number=Sing
S001	S002	Carta	NOUN	Gender=female	Number=Sing
S001	S003	Rector	NOUN	Gender=male	Number=Sing

(a)



(b)

Figure 2: (a) Adjacency table for entity-entity relationship (b) Graph representation for entities

The adjacency table represents the relationship between entities or words within a sentence. In the figure 2(b) the entity $Kast$ (i.e., a proper name) is connected to $Carta$ (letter) and $Rector$ (rector or principal), where columns $F1$ and $F2$ in the table are the dependency relations to be further assessed by the GA by using specific-purpose heuristics. For example, a *gender heuristic* will measure the agreement in gender between the entities being evaluated so that a high weight will be assigned to the connection between $Kast$ and $Rector$, whereas a low weight will be assigned to the connection between $Kast$ and $Carta$ as they differ in gender. Initially, every entity is connected with every possible entity and mentions within a sentence, and so constraints and heuristics will assign higher weight values to the connection between these entities as the GA-based optimization task goes on.

A graph is created from these adjacency tables as follows: Let G be a directed graph $G = G(V, E)$ where V is a set of vertices and E a set of edges, where each mention and entity within a text is represented as a vertex v and an edge e that is added to the graph for every pair of vertices representing mentions which can potentially refer to the same entity. In addition, a set of constraints between two mentions is used to assign a (connection) weight for each edge. Furthermore, let $X = (x_1, \ldots, x_n)$ be the set of candidate mentions to resolve for each feature x_i so that a vertex v_i is added to the graph with the corresponding features $x_i = (x_1 v_1, v_2 v_2, \ldots, x_n v_n)$. For example, a feature such as $x_1 v_1$ [23] can represent the 'gender' identified by the dependency parser.

3.4 Linguistic Heuristics and Constraints

The score of a candidate referents chain is computed by adding the scores of each voting heuristic and constraint features. A heuristic value aims to reward some features (i.e., positive score) whereas a constraint aims to avoid some linguistic features (i.e., negative score). Thus, an overall score for every candidate hypothesis is a function of its total number of 11 features: $Score_k = \sum_{i=1}^{i=11} x_{k_i}$.

Heuristics are first applied in order to assess and assign weights based on the linguistic behavior of each pair entity-reference. Weights are related to the strength with which candidate pronouns are a referent to a target entity, hence the score they assign will be assigned a positive weight. Overall, each pair will be assigned the total sum of weights obtained by each heuristic based on [16]:

1. *Definite Precedence (+0.2):* Nouns that are preceded by a demonstrative pronoun (or a definite determiner) have a higher chance of being antecedents of an entity.

2. *Not Prepositional Substantive Phrase (+0.1):* A noun phrase that occurs within a prepositional phrase is less likely to be the anaphora of the target entity.

3. *Pleonasm (+0.1):* One or more entities are redundant so that there are syntactic patterns of pleonastic anaphora which refer to the entity being evaluated.

4. *Syntactic Parallelism (+0.2):* Noun phrases are preferred with the same syntactic function as the anaphora .

5. *Semantic Parallelism (+0.1):* Noun phrases are preferred with the same semantic role as the anaphora .

6. *Recency (+0.2):* There is a higher chance of a candidate pronoun to be a referent of the target entity if this is in 'window' closer to the entity.

Constraints are then applied in order to weaken a relationship between entities that might be related. Hence constraints can vote negatively on each pair $(entity, pronoun)$ based on meeting the following selection restrictions and weights based on [6]:

1. *Gender Agreement (-0.2):* if the genders of the pair do not agree.
2. *Number Agreement (-0.2):* if the numbers (i.e., singular, plural) of the pair do not agree.
3. *Person Agreement (-0.1):* if the grammatical role of the pair do not agree.
4. *Reflexive Pronoun (-0.1):* if the pronoun does not refer to the subject/object of the clause (i.e., *themselves* versus *himself*).
5. *Semantic Agreement (-0.1):* if the semantics between anaphora and antecedent does not agree (i.e., they do not have the same logical connection provided by the dependency parser).

As a result of applying constraints and heuristics, a list of candidate entities and their corresponding scores is generated. Thus, for each sentence with ambiguous referents, every of its entities are treated as a node in the graph which is connected to every other entity in the sentence. Heuristics and constraints then 'vote' for candidate referents, and then the adjacency matrix is generated containing parsed sentences (i.e., relationships, linguistic features, POS tags). Next, the graph (figure 3) with the assigned weights is created for the recognized mentions and entities that being evaluated are represented.

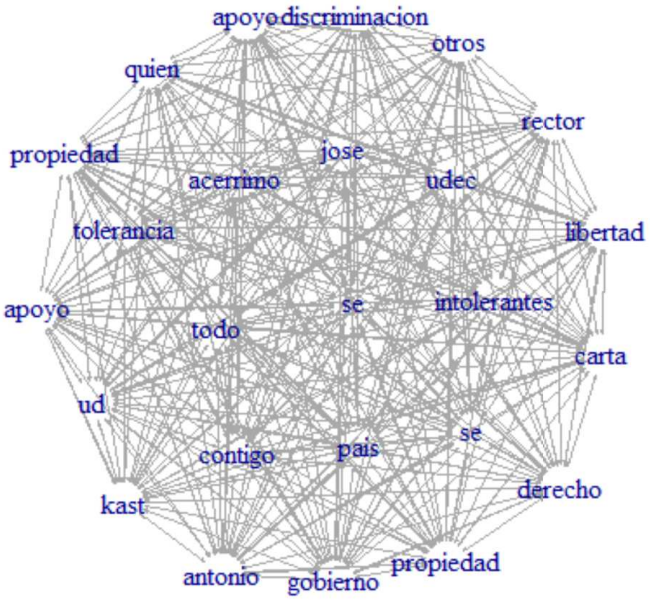


Figure 3: A sentence graph

The best anaphoric chains should be looked at by matching the annotated corpus. However, since there are too many candidate chains to search for, an

evolutionary computation method is responsible to explore and find the best co-reference chains.

3.5 Evolutionary Learning for Anaphora Chains

Searching and optimizing the best solutions is conducted by using a GA. Starting from an initial population of individuals (aka., hypotheses), a GA searches for the best solutions in a large search space by reproducing them and generating a better offspring as the evolution goes on based on the individual's fitness evaluation. In a GA, individuals can represent any set of hypotheses that may become the fittest ones based on a goodness evaluation as the GA goes on (figura 4).

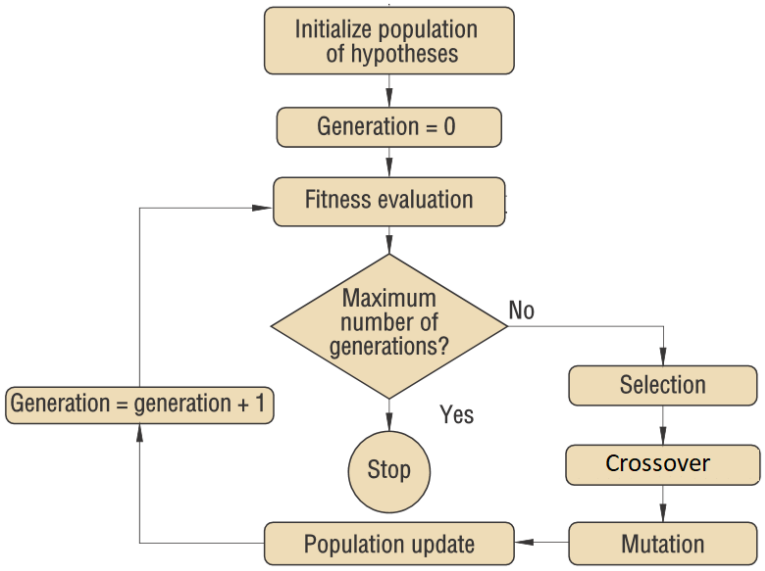


Figure 4: Structure of a Simple GA

Compared to other machine learning and optimization techniques, GAs perform global search by exploring solutions in parallel, and are robust to cope with noisy and missing data. In addition, they can search spaces of solutions containing complex interacting parts.

After creating an initial population of strings at random, genetic operations are applied with some probability in order to improve some individuals of the population. Once a new hypothesis is created, this is evaluated in terms of its measure of individual goodness referred to as *fitness*. Individuals for the next generation are selected according to their fitness values, which will determine those to be chosen for reproduction. If a termination condition is not satisfied, the population is modified by the operators and a new (and hopefully better)

population is created for each generation until more highly fit chromosomes are generated.

In our simple GA, three basic operations can be distinguished: population update (i.e., responsible for selecting the best individuals for reproduction), the genetic crossover and mutation operators (i.e., responsible for producing new offspring), and the fitness evaluation [1, 20]:

1. *Population Management:* it aims at selecting duplicates of good solutions for reproduction via the genetic operators, and at updating the population once the offspring have been produced. Selection mechanisms basically make duplicates of good solutions, while keeping the population size constant (by eliminating bad solutions in a population). This is usually achieved by identifying good solutions in the population, and then making multiple copies of good solutions. The selection of individuals may be implemented in a number of ways. Some common methods include tournament selection, elitism, proportional selection, and ranking selection.

2. *Crossover:* since selection cannot create new solutions in the population, a recombination operator, crossover, is introduced. In the simple case (i.e., single-point crossover), two individuals are picked from the mating pool (i.e., population) based on their fitness, and then some portions of these strings are exchanged to create two new individuals. Specifically, a single crossover position is chosen at random and the parts of two parents after the crossover position are exchanged to form two offspring.

3. *Mutation:* Even though crossover effectively searches and recombine individuals, occasionally it may become overzealous and lose some potentially useful genetic material. Hence a mutation operator is a random walk through the string space. When used with crossover, it avoids premature loss of important information. Usually, this operator is the occasional (with small probability) random alteration of the value of a string position. An improvement is not guaranteed during a GA generation, but it is expected that if bad strings are created, they will be removed by the selection method in subsequent generations and if good strings are created, they will be emphasized. In binary coding, this means changing the value of a 1 bit of a individual to 0 or viceversa. For our task, each gene in a chromosome is represented as a positive real value for a heuristic and a negative value for a constraint. Hence a mutation means incrementing or decreasing a random value for the current gene.

As previously mentioned, an individual is represented as a chromosome which is composed by real-valued genes representing features coding [23] its heuristics and constraints. Whenever a pair of parents is selected for crossover, they will share their genetic information for the generation of a hopefully better offspring that has a mix of genetic material of both parents. As an example of this operation, consider two parent individuals are selected for crossover as seen in figure 5.

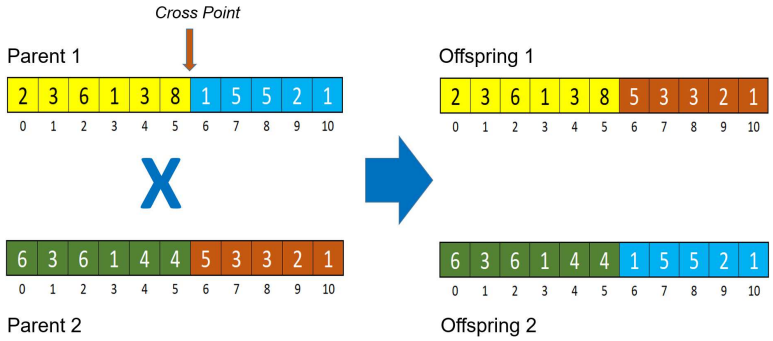


Figure 5: Single-point crossover between two individuals

Both parents share their genetic material by exchanging some portion of their genes at some given location. This means there is a range that can be chosen for the exchange to be conducted by randomly picking position 6 as a crossing point, the offspring would look like that shown at the right-hand size in figure 5. Some individuals are then picked up for mutation in which random changes are made on genes of the individuals so as to diversify the genetic pool and to avoid bias of the population and local optima.

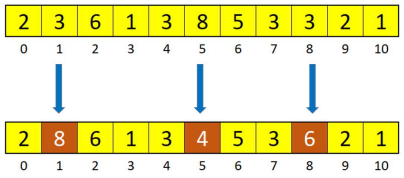


Figure 6: Simple real-valued mutation for a sample individual

For example, in the individual of figure 6, each gene is randomly chosen for mutation with some probability. In this case genes in positions 1, 5 and 8 are mutated by randomly adding/decreasing a real value to the gene's value, indicating some heuristic/constraint features that are changing.

For our approach, the GA uses single (random) permutation with probability P_m, simple crossover (with probability P_c), and a selection strategy (i.e., population update) based on elitism, this is, the best individuals of each generation will survive (i.e., they keep unchanged) to the next generation [1]. Each chromosome is represented as a string of 11 genes or features (i.e., 6 heuristics and 5 constraints) voting weight' values, so each position in the chromosome represents a heuristic/constraint from section 3.4, and the gene value represents how much the vote of that specific feature will weight in the total sum of votes for the individual.

For instance, for the chromosome of figure 7, heuristic values are represented

in the first 6 positions whereas constraint values are represented in the next 5 positions. Note that feature values assigned to each gene are based on selection restrictions proposed in [6]. Thus the heuristic at position 0 represents the 'definite precedence' from linguistic features in section 3.4 and has a (positive) 'voting weight' of 2, whereas the constraint at position 8 represents the 'person agreement"' feature and has a (negative) 'voting weight' value of 3, and so on with the rest of the values. Since the heuristic at position 0 is higher than the constraint at position 8, it has more impact when voting for a total score of a chromosome. Hence the overall score of a candidate solution is the sum of the positive values (i.e., heuristics) and the negative values (i.e., constraints) which gives always a positive score. Once the GA converges, there will be a set of assessed chromosomes that represent the best combination of heuristic and constraints 'voting weights' for a referent/entity relationship.

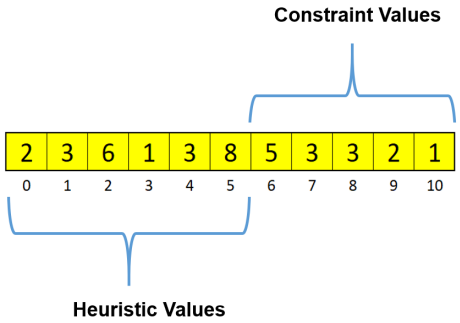


Figure 7: Gene representation for linguistic heuristics and constraints.

In order to assess every individual (aka. fitness evaluation) as the GA goes on, a fitness function evaluates the proportion of correctly resolved referents over the reference annotated corpus by using the set of solutions created by the GA: $F(X) = \sum IsMention(X, m)$, where $IsMention(X, m)$ is 1 if the method correctly selects the candidate mention m using the chromosome X, and 0 otherwise. For example, for the Spanish sentence `'El presidente escribió una carta, es terrible.'` (i.e., *'The president wrote a letter, it is terrible'*), the graph in figure 8 is generated. Relationships between entities and mentions are to be improved as the GA goes on by recombining this graph-based chromosome with other hypotheses with the hope of producing the best candidate solutions, this is, those having the best voting weights for heuristics and constraints.

4 Experiments and Results

In order to assess the performance of our GA-based co-referencing method, a prototype was implemented by using the NLP libraries in Python and R. In addition, key tasks were conducted including text collection and pre-processing,

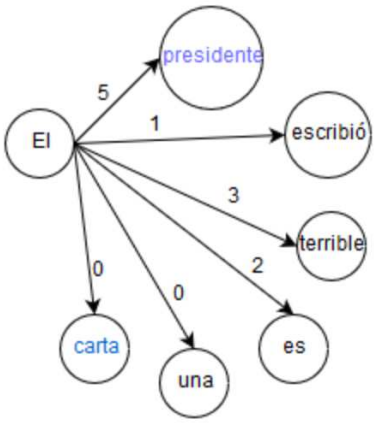


Figure 8: Graph representation for a single sentence.

graph representation, and GA implementation, as follows:

4.1 Text Collection and Preprocessing

In order to create a corpus of short text messages, the twitter API (https://developer.twitter.com/en/docs/tweet reference/get-search-tweets) was used to collect conversation threads, in which every *tweet* is seen as a standalone sentence. Threads were used as context for referencing entities so that nouns and pronouns could be more easily identified when referencing is manually made.

Overall, for a given topic, 4525 *tweets* were downloaded as a tree-like structure whose content is based mainly on local news since 2018 in which an 80% was used for training and 20% for testing purposes. In addition, the corpus was manually annotated with co-references (i.e., chains of entity and mentions) so as to assist the GA learning.

Linguistic pre-processing tasks were then performed including *POS tagging* (i.e., $CoreNLP$ and $OpenNLP$ packages), *Cleaning* (i.e., R library *stringr*), NER (i.e, NLTK and SpaCy Python libraries), *Dependency parsing* (i.e., $Udpipe$ R package).

4.2 Graph Representation

Once the corpus has been pre-processed, a graph-based representation is created for every annotated sentence. Adjacency matrices containing POS tags and dependency relations are created by using regular expressions and the *igraph* R package.

4.3 GA Implementation

Performance evaluation of the GA (i.e., fitness) was conducted by adjusting different parameters such as probability of crossover (P_c) and probability of mutation (P_m), size of the annotated sentences (i.e., corpus) and number of generations. Thus, graphics of figures 9 and 10 show the evolution of the fitness based on incremental values for P_c and P_m.

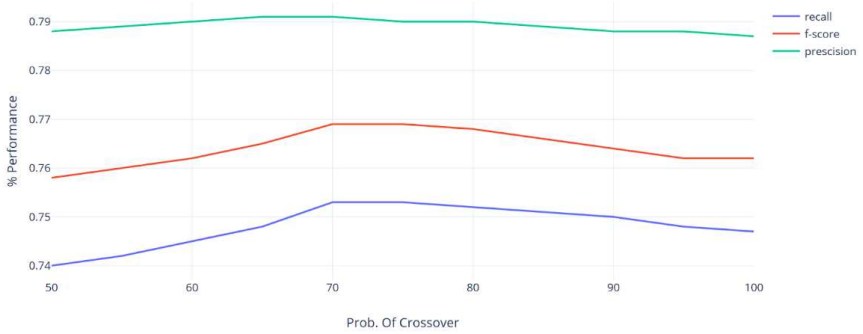


Figure 9: Performance vs P_c

Based on the best experimental settings, the final GA's initial population was randomly created with 50 individuals with a maximum of 1000 generations, $P_m = 0.01$, $P_c = 0.7$ and an elitist strategy for the population update. The GA was run until relatively stable fitness evaluations were obtained (figure 11).

Final co-reference chains were extracted by comparing the relationships between mentions and entities provided by the GA (the best offspring) and the co-reference annotated corpus (baseline). Accordingly, correctly detected referents were computed achieving a $Precision = 0.80$, $Recall = 0.75$ and $F1 = 0.77$. Additional assessments were conducted by measuring $F1 - Score$ for different corpus sizes in order to investigate the extent to which the overall performance may be dependent on the available corpus (figure 12).

Results show the model is relatively independent on the annotated corpus size as it manages every sentence in a very atomic way. Notice that the drop of performance when some chains are not found on the annotated corpus is not significant, suggesting the approach may achieve fair results without using a large corpus.

Furthermore, comparisons with some state-of-the-art approaches can be highlighted in table 1. However, note that most of them are intended for processing full-sized and formal natural-language texts using large training corpus. Results show our approach ranks in a very competitive position for informal and short message co-reference resolution.

Unlike other methods, experiments showed our approach is not strongly dependent on large training corpus as long a 'thread structure' is provided from a micro-blogging system. In addition, the majority of the approaches use formal

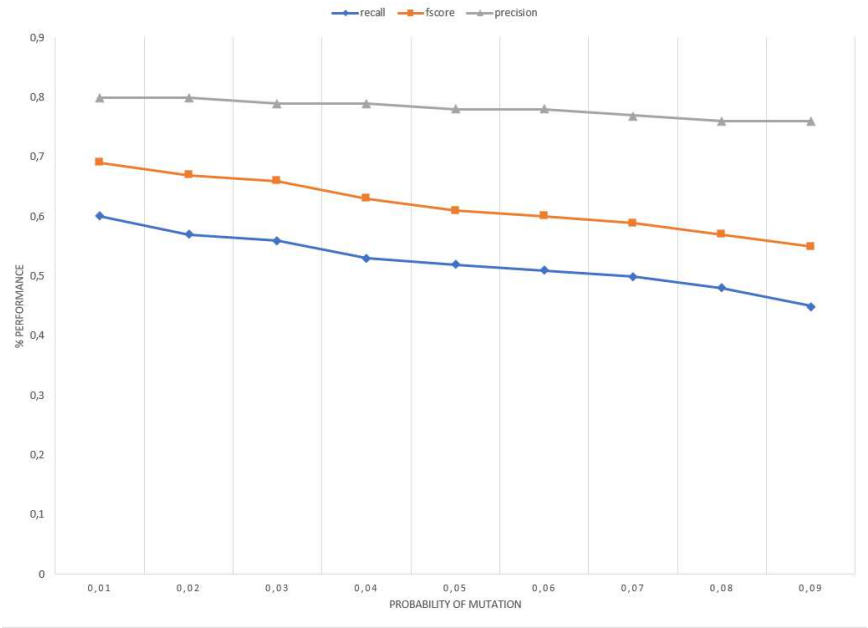


Figure 10: Performance vs P_m

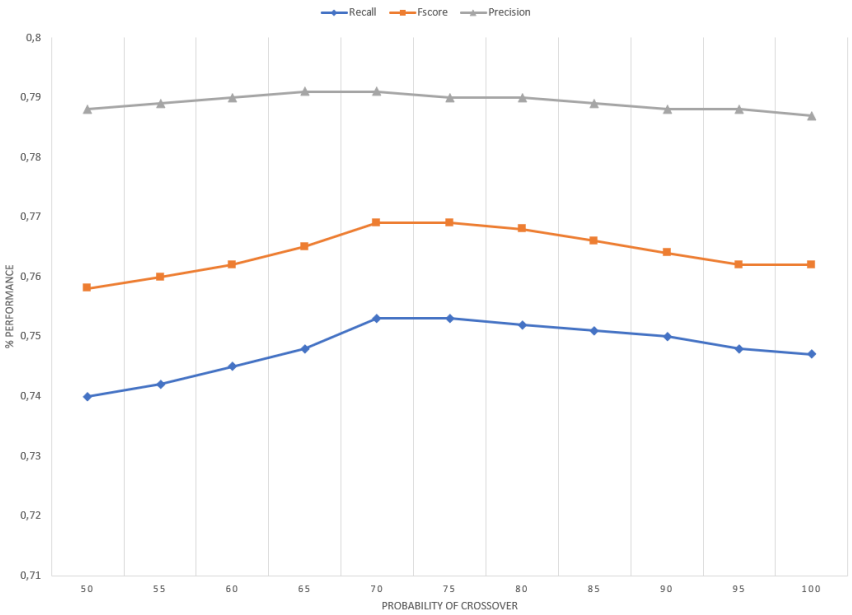


Figure 11: Fitness evolution for the best solutions.

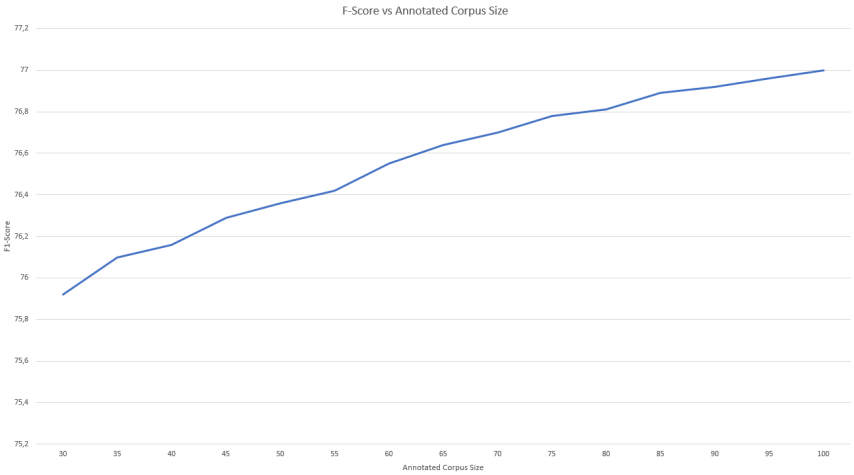


Figure 12: F1-Score vs Annotated Corpus Size

Method	**Type**	**Size of Sample**	**Performance**
Hobbs Naive (baseline) [11]	Formal computer manuals, nursing notes, Wall Street Journal articles	3900 sentences from the Treebank corpus	$F1 = 67.8\%$
Naive Classifier [6]	Formal Hand-crafted dialogues for an educational environment	Some hundreds of sentences	$F1 = 83\%$ and Avg. Accuracy $= 65 - 80\%$
Unsupervised method on the Ontonotes dataset [2]	Informal short messages and news	MUC-7 (58.594 words)	Precision=0.74, Recall=62.45
Memory-based Learning [3]	Informal short social media messages	Testing involved 390 tweet messages and 370.000 sentence instances (ANCORA corpus)	Precision $= 0.8 - 0.92$ Recall $= 0.62 - 0.8$
Our GA-based approach	Informal short social media messages	Training involved 4,525 tweets (24,090 words)	Precision $= 0.80$, Recall $= 0.75$ and $F1 = 0.77$

Table 1: Comparing state-of-the-art approaches.

and relatively large corpus so it is unclear whether they can perform well for different kind of texts.

Some current supervised ML methods for classification (i.e., word embedding models) are extremely dependent on large training corpus as they rely on analyzing context terms that surround target words on specific 'windows', whereas for twitter messages, context windows for training are quite short and most of the time do not contain useful information to make further inferences. Nevertheless, our model's performance may be increased by using more specific linguistic heuristics and constraints as dealing with informal and short texts prevents methods from using effective discourse processing techniques usually seen in full natural language texts.

5 Conclusions

In this work, a GA-based model for automatic co-reference resolution for microblogging (i.e., twitter) applications was presented. The approach addresses several issues found in social media messaging such as informal texts, short messages, slangs. In order to evolve toward the best solutions (aka. the best linguistic co-referencing chains connecting entities and referents) specific-purpose genetic operators on a graph representation are applied and linguistic filters consisting of constraints and heuristic are used to compute the fitness of every candidate solution. The graph becomes an efficient representation of the weighted connections in a co-referencing chain. This plays a key role in social media as text messages usually assume that opinions containing explicit references to entities/features are not recognized, hence much opinion messages can be missed by opinion retrieval tasks for further applications such as sentiment analysis, opinion mining.

Our approach was assessed by using informal text corpus, achieving competitive results against state-of-the-art methods but it does not require large amounts of annotated corpus to produce fair solutions. Furthermore, limitations on the length of messages on platforms such as *twitter* restrict users from expressing richer linguistic information and from providing multiple referential chains within a tweet. Despite this, training a model on annotated informal corpus showed very promising results in terms of a higher classification accuracy for detecting referential chains.

References

[1] M. Affenzeller, S. Winkler, S. Wagner, and A. Beham. *Genetic Algorithms and Genetic Programming: Modern Concepts and Practical Applications*. Chapman and Hall/CRC, 2009.

[2] Berfin Aktaş, Veronika Solopova, Annalena Kohnert, and Manfred Stede. Adapting coreference resolution to twitter conversations. *Association for*

Computational Linguistics: EMNLP 2020, November, page 2454–2460, 2020.

[3] J. Atkinson, G. Salas, and A. Figueroa. Improving opinion retrieval in social media by combining features-based coreferencing and memory-based learning. *Information Sciences*, 299(1):20–31, 2015.

[4] S. Bowman and E. Pavlick. Looking for elmo's friends: Sentence-level pretraining beyond language modeling. *ICLR*, 2018.

[5] D. Byron and J. Allen. Applying genetic algorithms to pronoun resolution. *AAAI '99/IAAI '99 Proceedings of the sixteenth national conference on artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, page 957, July 2000.

[6] J. Cai and M. Strube. Evaluation metrics for end-to-end coreference resolution systems. *SIGDIAL '10 Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 28–36, September 2010.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, 2019.

[8] A. Emami, P. Trichelair, A. Trischler, and K. Suleman. The knowref coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution. *ACL*, page 9, June 2019.

[9] Y. Goldberg and G. Hirst. Neural network methods in natural language processing. *ArXiv*, 2017.

[10] Iris Hendrickx and Veronique Hoste. Coreference resolution on blogs and commented newsanaphora processing and applications. *7th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC, Goa, India, November*, pages 43–54, 2009.

[11] Yan Huang. *Anaphora: A cross-linguistic approach*. Oxford University Press, 2000.

[12] G. Kim, L. Schubert, and A. Type-coherent. A type-coherent, expressive representation as an initial step to language understanding. *Proceedings of the 13th International Conference on Computational Semantics , Association for Computational Linguistics*, pages 13–30, May 2019.

[13] F. Liu, L. Zettlemoyer, and J. Eisenstein. The referencial reader: A recurrent entity network for anaphora resolution. *57th Annual Meeting of the Association for Computational Linguistics (ACL)*, July 2019.

[14] Y. Liu, M. Gardner, and M. Lapata. Structured alignment networks for matching sentences. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium*, page 1554—1564, 2018.

[15] Xiaoqiang Luo, Radu Florian, and Todd Ward. Improving coreference resolution by using conversational metadata. *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, June*, pages 201–204, 2009.

[16] R. Mitkov. Anaphora resolution: The state of the art. In *ACL*, pages 1–34, 2007.

[17] C. Muthuraman, Y. Yinfei, D. Cer, S. Yuan, Y. Sung, B. Strope, and R. Kurzweil. Learning cross-lingual sentence representations via a multi-task dual-encoder model. *Repl4NLP, July*, 2019.

[18] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July*, pages 104–111, 2002.

[19] E. Sapena, L. Padró, and J. Turmo. A graph partitioning approach of coreference resolution. Technical Report LSI-09-2-R, LSI, UPC, Spain, January 2009.

[20] C. Sheppard. *Genetic Algorithms with Python*. CreateSpace Independent Publishing Platform, 2016.

[21] L. Xiaoqiang. On coreference performance metrics. *HLT Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 25–32, October 2005.

[22] Y. Xu and J. Yang. Look again at the syntax: Relational graph convolutional network for gendered ambiguous pronoun resolution. *ACL 2019 Workshop on Gender Bias for Natural Language*, May 2019.

[23] B. Xue, M. Zhang, W. N. Browne, and X. Yao. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation*, 20(4):606–626, 2016.

[24] H. Zhang, Y. Song, and Y. Song. Incorporating context and external knowledge for pronoun coreference resolution. *CoRR*, abs/1905.10238, 2019.