International Journal of Information Management Data Insights 2 (2022) 100115

Contents lists available at ScienceDirect

International Journal of Information Management Data Insights

journal homepage: www.elsevier.com/locate/jjimei

Evolutionary natural-language coreference resolution for sentiment analysis

John Atkinson[a,*], Alex Escudero[b]

[a] Universidad Adolfo Ibañez, Santiago, Chile
[b] Whiz, Peru

ARTICLE INFO

Keywords:
Natural-language processing
Sentiment analysis
Coreference resolution
Information management
Social media analysis
Evolutionary computation

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 coreference resolution task involves detecting coreference chains of entities and pronouns that coreference them. It 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

In today's world, the amount of stored information has been enormously increasing in the unstructured form and cannot be used for any processing to extract useful information, so unstructured data analysis techniques known as text analytics must be applied. These include summarization, classification, clustering, information extraction and visualization. Hence text analytics aims at extracting interesting and non-trivial patterns or knowledge from texts aiming to discover insights with very high commercial values (Shao, Hua, & Song, 2021; Sharma, Rana, & Nunkoo, 2021a).

A huge source of textual information such as social media has created unprecedented opportunities for people to publicly voice their opinions, but has created serious bottlenecks when it comes to making sense of these opinions (Yerasani, Tripathi, Sarma, & Tiwari, 2020). The Web contains a wealth of opinions or text messages about products, politicians, and more, which are expressed in newsgroup posts, review sites, and elsewhere (Henkel, Perjons, & Sneiders, 2017; Martin, Rice, & Arthur, 2020; Noori, 2021). As a result, the task of opinion mining or sentiment analysis has seen increasing attention over the recent years (Georgiadou, Angelopoulos, & Drake, 2020; Jabar, Alkhatib, & Wannous, 2022), with plenty of techniques to deal with this. In order to mine for sentiments on opinion messages, one must first be capable of retrieving relevant opinions from social media so as to identify information such as the opinion's polarity and target object (i.e., Which part is opinion expressing), the holder (Who wrote the opinion), etc. A key problem is that information retrieval engines on opinions assume that target objects and their features are explicitly contained in a text. However, texts on social media (Chen, Davison, & Ou, 2020; Hajli & Featherman, 2017) contain a lot of implicit knowledge on the target topic of the sentiment itself. For example, in twitter a text's main object is usually referred from sub-comments but semantically related (Sharma, Rana, & Kumar, 2021b). It may cause that opinions containing different objects (but conceptually equivalent) are seen as different ones between each other due to the implicit way different entities/objects can be referenced which may lead to a significant drop in the further sentiment classifiers' accuracy.

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. Consequently, since the task fails to understand the synonym relationship, the quality of

[*] Corresponding author.
E-mail addresses: john.atkinson@uai.cl (J. Atkinson), alex.escudero@whiz.one (A. Escudero).

https://doi.org/10.1016/j.jjimei.2022.100115
Received 16 November 2021; Received in revised form 20 August 2022; Accepted 21 August 2022

a further application such as sentiment analysis (Bessou & Aberkane, 2019; Darwich, Azman, Noah, Omar, & Osman, 2019) 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. In Natural-Language Processing (NLP) and information management (Naredla & Adedoyin, 2022), this is a key discourse analysis process which is usually carried out on formal full natural-language texts (Palivela, 2021). However, no significant evidence has been found to effectively perform this task from short and informal text messages lacking much linguistic knowledge.

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 (Palivela, 2021). 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, *Coreference Resolution* (CR). This effort is heavily used in tasks such as sentiment analysis, information management, and information extraction (Shao et al., 2021; Yufang, 2020). CR techniques (Atkinson, 2022; Sukthanker, Poria, Cambria, & Thirunavukarasu, 2020) 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 the state-of-the-art (SOTA) to enable applications using informal text data such as text categorization, lexical tagging and named-entity recognition (Kumar, Kar, & Ilavarasan, 2021; Sukthanker et al., 2020).

Accordingly, in this paper a new computation model for discovering language co-references in short and informal texts is proposed for sentiment analysis applications. It combines evolutionary learning and graph-based representations in order to detect and extract implicit knowledge (aka. co-reference chain). It uses linguistic context and a weights system that are assessed by a domain-dependent heuristic in order for an evolutionary computation method to discover co-reference chains.

2. Related work

Opinion retrieval is a document retrieving and ranking process. A relevant document must be relevant to a query and contain opinions toward the query. Opinion polarity classification is an extension of opinion retrieval, which classifies the retrieved text as positive, negative or mixed, according to the overall polarity of the query relevant opinions in the document. Overall, opinion retrieval is a computational task that identifies subjective information from human-generated textual contents in order to retrieve textual information that contains sentiments (Ahmad & Laroche, 2017; Atkinson, 2022; Jabar et al., 2022). Further inference on opinion retrieval describes extraction and judgment analysis (Fujii, Sakaji, Masuyama, & Sasaki, 2022; Wang & Yu, 2017) on various aspects of opinionated contents. However, the arrival of social media creates an avenue whereby humans show their diversities in terms of writing styles and language uses.

These challenges have made opinion retrieval task difficult. It has also led to active research on opinion retrieval and sentiment analysis (Darwich et al., 2019), and had since recorded a huge number of research works (Wadud et al., 2022). Some of the key challenges can be attributed to the different ways of expressing opinions or sentiments by social media's users (Bessou & Aberkane, 2019). Other important usage of opinion retrieval include, analysis of questionnaire or survey responses, personalized search engines, opinion or sentiment analytic, and understanding consumers sentiments from products or movie reviews (Noori, 2021).

For language expressed in opinion texts, there is an underlying link between a target object (antecedent) and its reference(s) in information available on social media, a kind of specific referencing linguistic problem known as anaphora, in which an antecedent can usually be found somewhere before this referential expression. Unlike the previous linguistic type of linking, lexical co-referencing determines Noun Phrases (NP) co-occurring in a discourse by generating a referencing chain, in which all the expressions refers to the same entity. For example, in the following text: *The bank X will keep its interest rates next year, so that its customers are not harmed. The financial entity is among the best ten...*, the co-referencing chain will contain the expressions *(The bank X, its, The financial entity)* as they refer to same object. Thus, co-reference analysis aims to detect chains of nouns or pronouns with a single referent within texts.

CR is a complex computational task as it must explore a search space involving a huge number of candidate antecedents (e.g., noun phrases containing target entities) and the referential expression with the outcome being a referencing chain. Traditional computational linguistic methods for solving anaphora use a full natural-language text as discourse knowledge and range from classical heuristic-based approaches to machine-learning co-referencing techniques such as support vector machines (SVM) and Neural Networks. To this end, CR methods can usually model a text's discourse structure in order to further look for coreference or anaphoric links (aka. chains). However, there is no such a clear discourse structure within short texts found in micro-blogging systems (i.e., twitter), e-commerce reviews, so SOTA 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 a text opinion. 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 short length of the texts restricts the task from performing deep linguistic processing.

Yet, well-known CR 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`. Furthermore, English pronouns tell us a lot about an entity being referred (i.e., `it, their, her, his`) whereas for other languages such as 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 CR approaches used syntactic and semantic heuristics (Sharma et al., 2021a) 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 (Cai & Strube, 2010). 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 (Cai & Strube, 2010). 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 (Batra, Jain, Tikkiwal, & Chakraborty, 2021). 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 conduct coreference resolution (CR) on formal language sentences extracted from the Treebank corpus.[1] 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.

Supervised ML techniques such as decision trees were also explored to pairwise classification of pairs of entity mentions by using graph inference methods. Once they are classified, single-link clustering is performed to produce final co-referential chains. This graph-based approach for CR (Emami, Trichelair, Trischler, & Suleman, 2019) 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 (Zhang, Song, & Song, 2019).

The task is then conducted by using probabilistic and deterministic learning algorithms such as *Relaxation Labeling* on different partitions of the graph (Emami et al., 2019). 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[2] which is composed of broadcast news (Nasir, Khan, & Varlamis, 2021), 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* (Atkinson, Salas, & Figueroa, 2015). The MBL-based classifier is trained on the ANCORA corpus[3] by using pairs of entities and referents, and takes into account the repetition of entities lying in the classical forum thread hierarchies, achieving an F-score = 0.74 for the CR task.

Further improvements of these methods by using the Ontonotes[4] training corpus on twitter conversations and the MUC-7 dataset (Aktaş, Solopova, Kohnert, & Stede, 2020) 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 (Hendrickx & Hoste, 2009; Martarelli & Nagano, 2022).

3. A GA-based graph approach for learning coreference chains

In this research, a new computational approach is proposed to address the CR task for social media texts by combining graph representation, genetic algorithms and heursistic-based metrics. It aims at feeding further sentiment analysis tasks.

Our approach uses messages' hierarchies (aka. thread structure) provided by twitter to identify two types of entities that feed our CR method:

1. *Original Messages:* they contain an answer posted by other users so that they can be located either at the root of hierarchy (i.e., the head) or in the middle.
2. *Reply Messages:* they are answers to previous messages (i.e., subcomments) and they usually do not contain explicit entities as these are first used by the original user. Instead, these messages contain referential expressions. Furthermore, a message can be original and reply at the same time.

The thread structure does not only contain referential expressions within the message but debate threads created from the original posted messages in the social network. Since there is poor linguistic information on a short message, related comments in a discussion provide us with 'hints' on the driving threads containing entities or features, which can also support the CR task. Accordingly, our approach can resolve co-references chains within the retrieved hierarchies. For example, two sample hierarchies can be seen in Fig. 2: that built up from messages (1)–(3), and that built up from messages (1)–(4). The working model is then based on three major tasks as seen in Fig. 1:

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 a set of candidate entities when solving coreferences and anaphora.

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

Our model can be seen in Fig. 2 in which a tweets dataset is collected from a given input query based on local news (Nasir et al., 2021), which are manually annotated for coreference 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 Genetic Algorithm (GA) then iteratively searches for and optimizes the best graphs representing coreference chains based on the reference annotated corpus and defined linguistic heuristics (Zhang, Sun, & Jara, 2015).

In order to collect a working corpus, the twitter API was used to download tweet threads on a given query based on local news. These are arranged as a tree hierarchy 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.

3.1. Text preprocessing

An important task in our model (Fig. 2) involves collecting the previously created twitter's threads and clean them up to remove specific-purpose 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 (Allam, Bliemel, Spiteri, Blustein, & Ali-Hassan, 2019), 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

[1] https://catalog.ldc.upenn.edu/LDC99T42
[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace
[3] http://clic.ub.edu/corpus/es/ancora
[4] https://catalog.ldc.upenn.edu/LDC2013T19

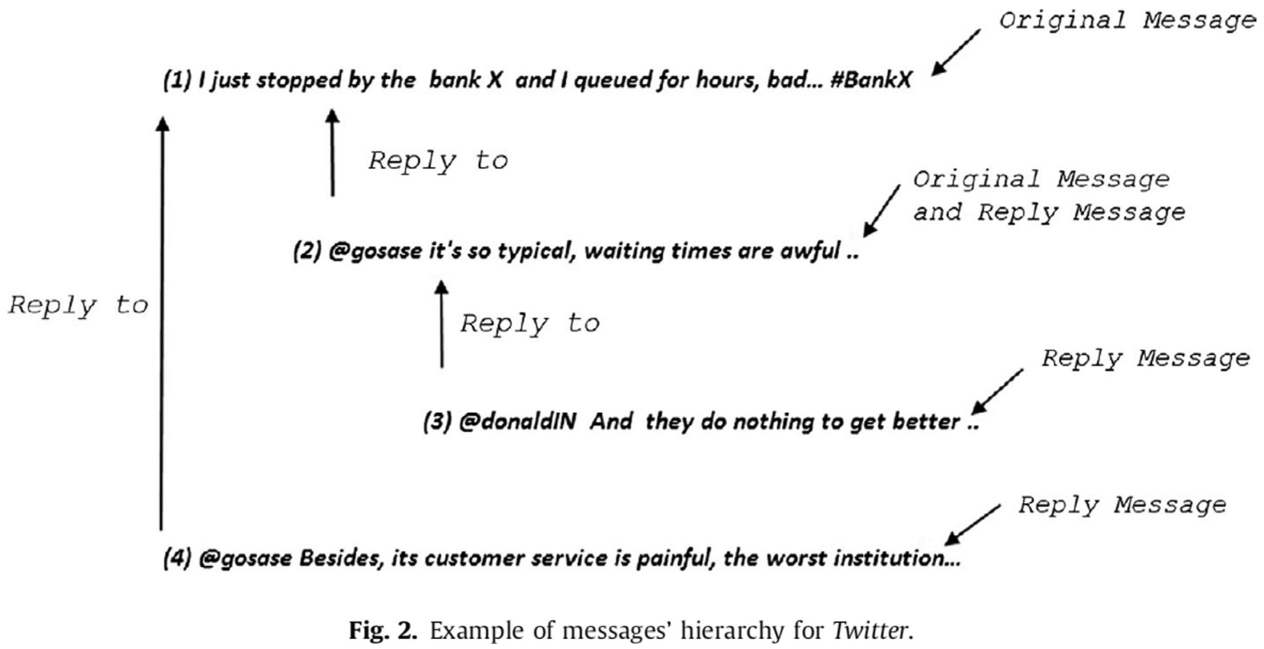

Original Message
(1) I just stopped by the bank X and I queued for hours, bad... #BankX
Reply to
Original Message and Reply Message
(2) @gosase it's so typical, waiting times are awful ..
Reply to
Reply to
Reply Message
(3) @donaldIN And they do nothing to get better ..
Reply Message
(4) @gosase Besides, its customer service is painful, the worst institution...
Fig. 2. Example of messages' hierarchy for Twitter.

Fig. 1. Example of messages' hierarchy for Twitter.

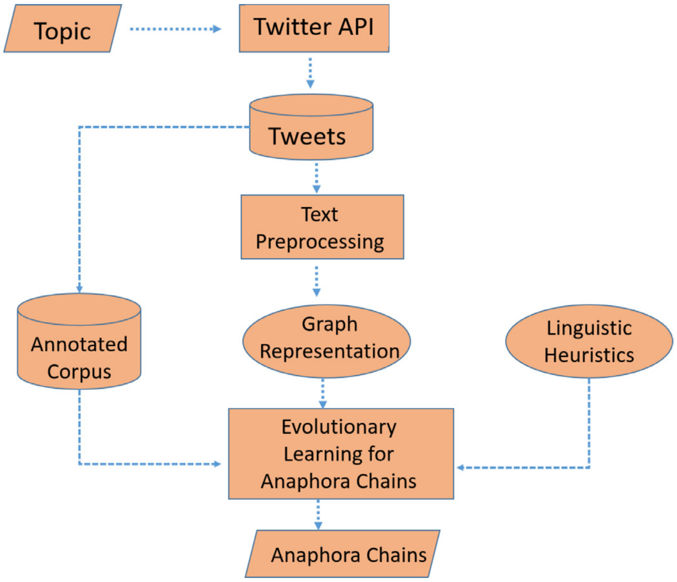


Fig. 2. Our Twitter-based evolutionary model for coreference resolution.

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, *Named-Entity Recognition* (NER) is applied to our annotated corpus to recognize useful multi-word entities such as organizations, people, locations, from lexical information provided by the POS tagger (Derczynski et al., 2015). 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 coreference chains.

3.2. Graph representation

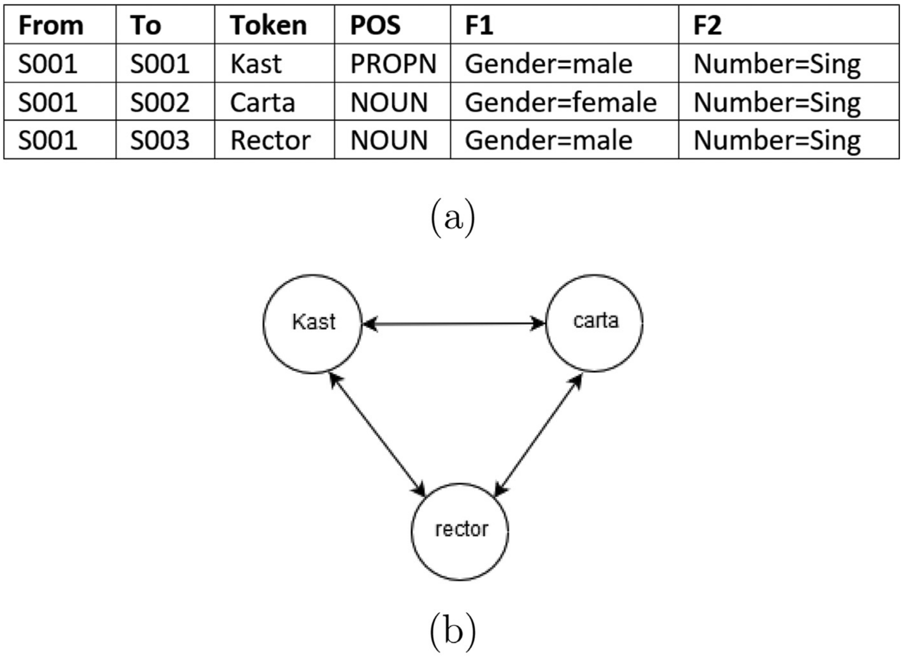
Once twitter entities and relationships have been identified, an adjacency table (Fig. 3(a)) is generated for every pair of sentences and then the corresponding graph representation (Fig. 3(b)) is produced from dependency relations.

The adjacency table represents the relationship between entities or words within a sentence. In the Fig. 3(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 $x_1 v_1$ (Xue, Zhang, Browne, & Yao, 2016) can represent the 'gender' identified by the dependency parser.

3.3. 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



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)

Fig. 3. (a) Adjacency table for entity-entity relationship (b) Graph representation for entities.

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:

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 (Cai & Strube, 2010):

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).

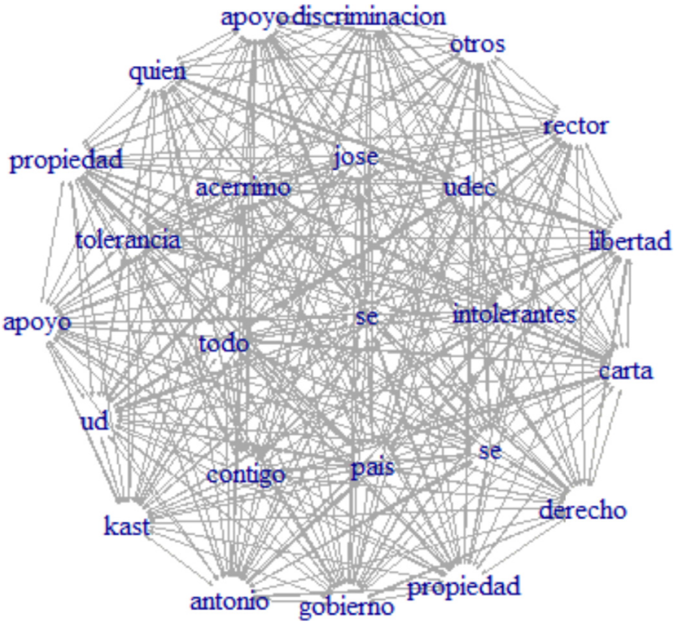


Fig. 4. A sentence graph.

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 (Fig. 4) with the assigned weights is created for the recognized mentions and entities that being evaluated are represented.

The best coreference chains should be looked at by matching the annotated corpus. However, since there are too many candidate chains to search for, an evolutionary computation (the GA) method is responsible to explore and find the best coreference chains.

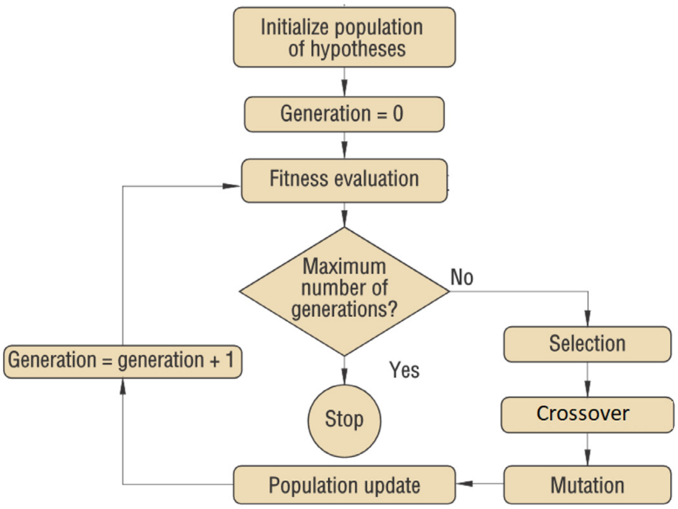

Initialize population of hypotheses
Generation = 0
Fitness evaluation
Maximum number of generations?
No
Yes
Selection
Crossover
Mutation
Stop
Population update
Generation = generation + 1

Fig. 5. Structure of a simple GA.

3.4. Evolutionary learning of coreference 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 (Fig. 5).

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 (Gupta, Kar, Baabdullah, & Al-Khowaiter, 2018). In addition, they can search spaces of solutions containing complex interacting parts (Lebib, Mellah, & Drias, 2017).

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 (Lebib et al., 2017; Sheppard, 2016):

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 parent individuals after the crossover position are exchanged to form two offspring. Since chromosomes (i.e., individuals) represent candidate coreference chains expressed as linguistic feature constraints, crossover simply exchanges linguistic features so a coreference chain gets better as the GA goes on.
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 represents a feature constraint for a candidate coreference chain.

As previously mentioned, an individual is represented as a chromosome which is composed by real-valued genes representing features coding (Xue et al., 2016) their 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 Fig. 6.

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 Fig. 6. Some individuals are then picked up for mutation in which random changes are made on genes of the individuals to diversify the genetic pool and to avoid bias of the population and local optima.

For example, in the individual of Fig. 7, 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. 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 3.3, 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 Fig. 8, 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 Cai & Strube (2010). Thus the heuristic at position 0 represents the 'definite precedence' from linguistic features in Section 3.3 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.

In order to assess every individual (aka. fitness evaluation) as the GA goes on, a fitness function evaluates the proportion of correctly

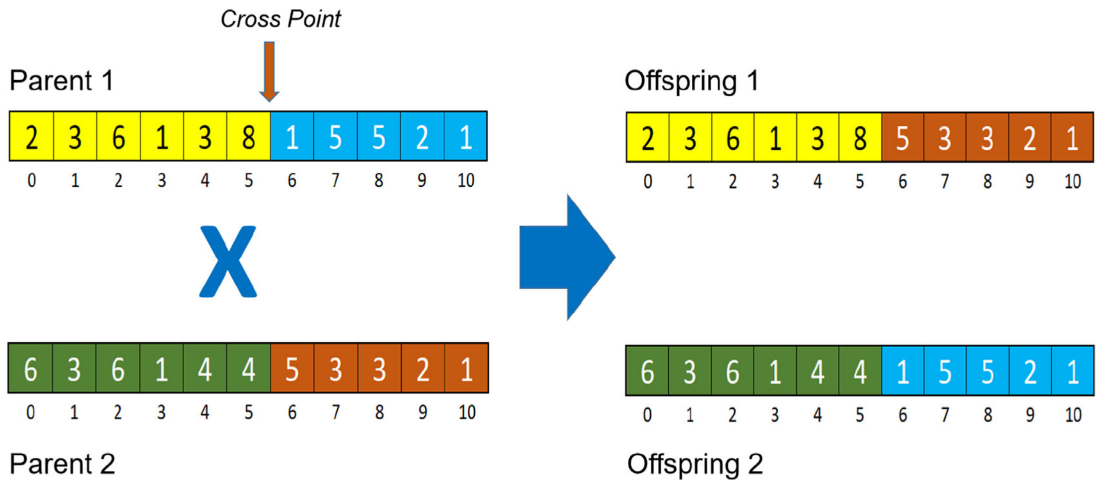


Fig. 6. Single-point crossover between two individuals.

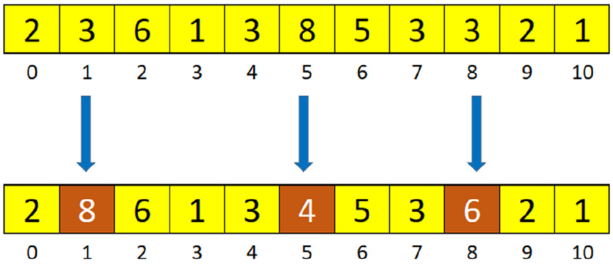


Fig. 7. Simple real-valued mutation for a sample individual.

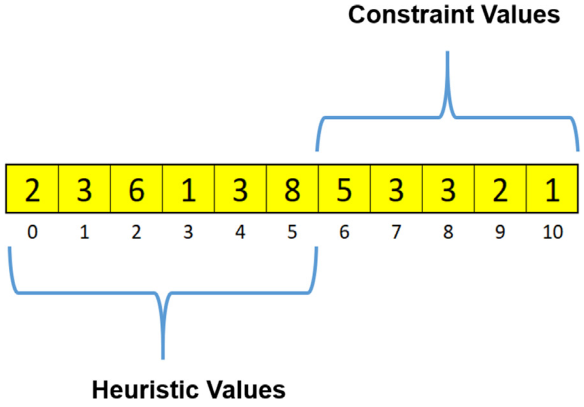


Fig. 8. Gene representation for linguistic heuristics and constraints.

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 Fig. 9 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. Evaluation

In order to assess the performance of our GA-based CR method, a prototype was implemented by using the NLP libraries in Python and R.

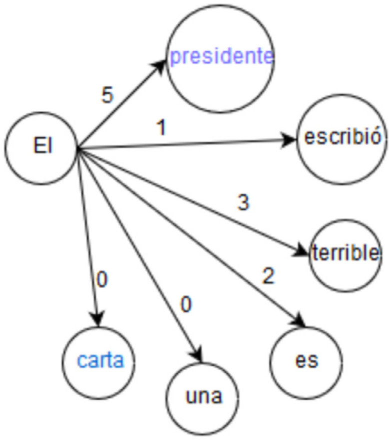


Fig. 9. Graph representation for a single sentence.

In addition, key tasks were conducted including text collection and preprocessing, graph representation, and GA implementation, as follows:

4.1. Data collection and preprocessing

In order to create a corpus of short text messages, the twitter API[5] 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 query, 4525 *tweets* were downloaded as a tree-like structure whose content was mainly built from local news randomly retrieved by using the above-mentioned API. It includes news from 2018 in which an 80% was used for training and 20% for testing purposes. In addition, the news collection was manually annotated with co-references (i.e., chains of entity and mentions) to assist the GA learning.

Linguistic pre-processing tasks were then performed including *Part-of-Speech (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).

Once the corpus has been pre-processed, a **graph-based representation** is created for every annotated sentence. Adjacency matrices con-

[5] https://developer.twitter.com/en/docs/tweets/search/api-reference/get-search-tweets

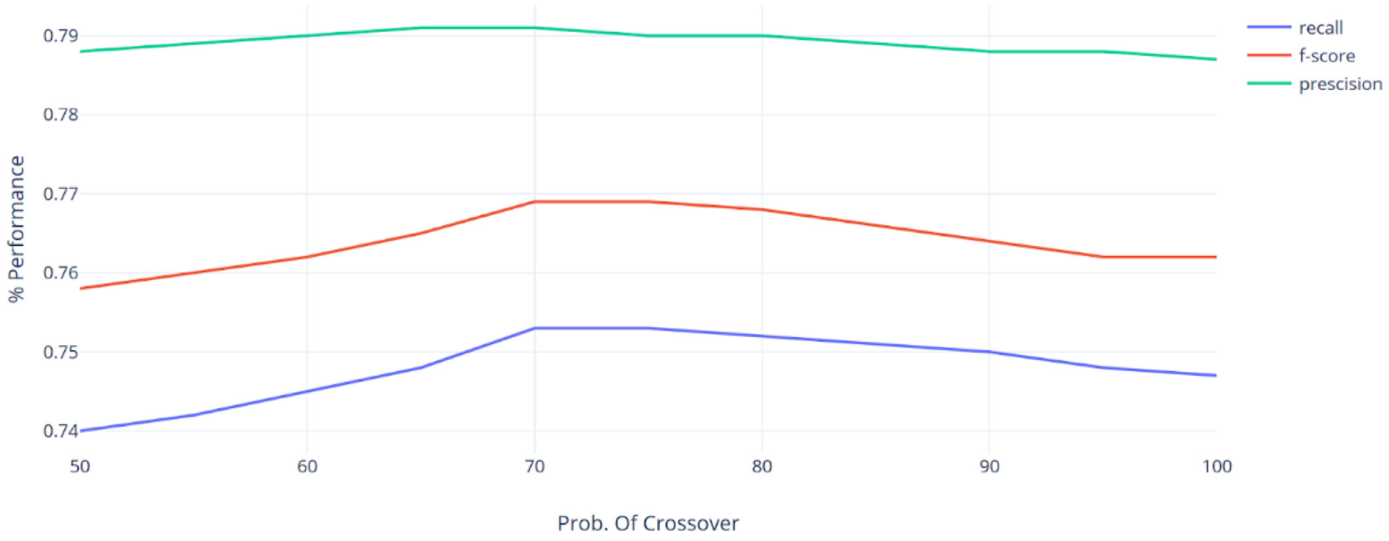


Fig. 10. Performance vs. P_c.

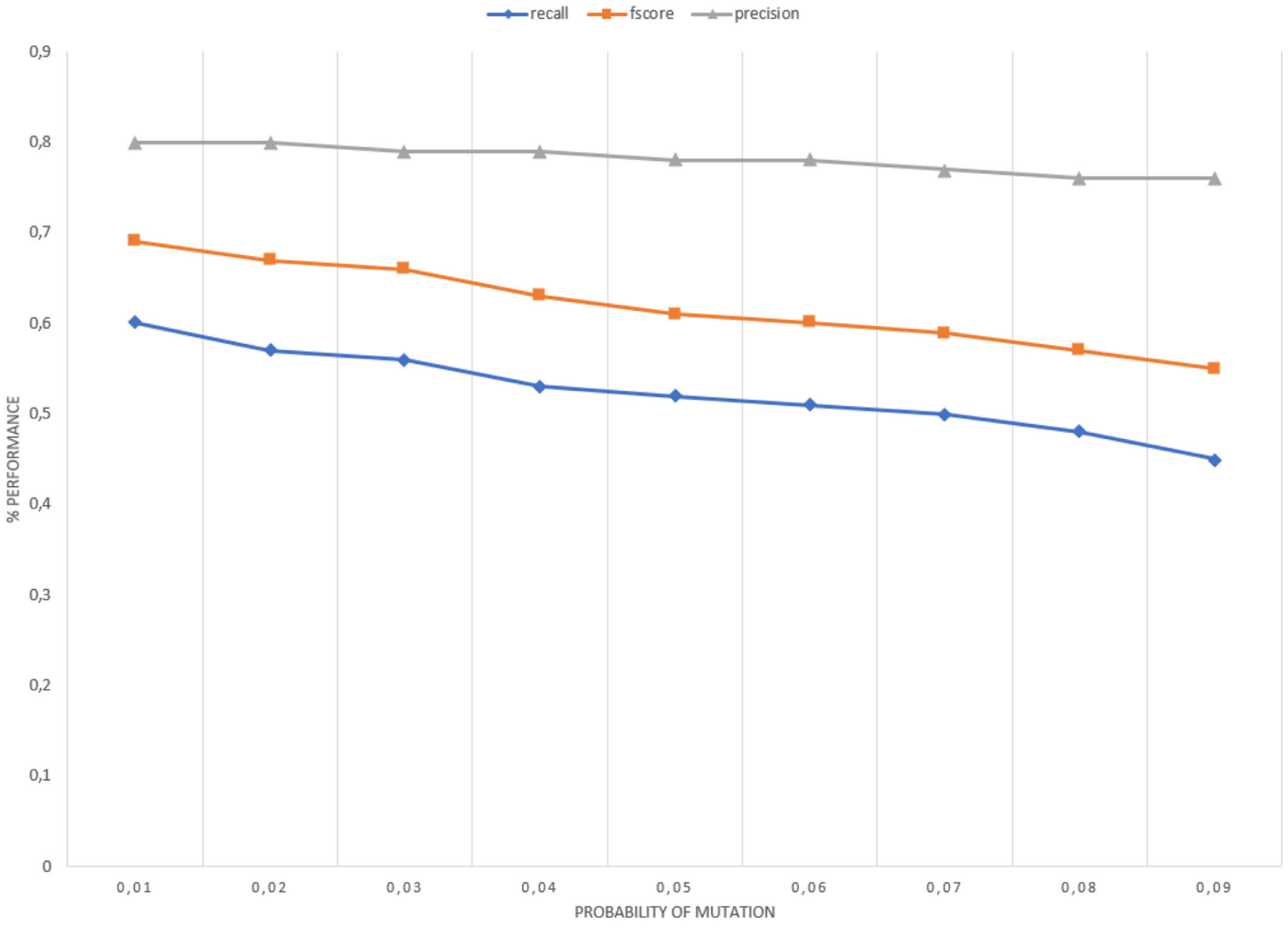


Fig. 11. Performance vs. P_m.

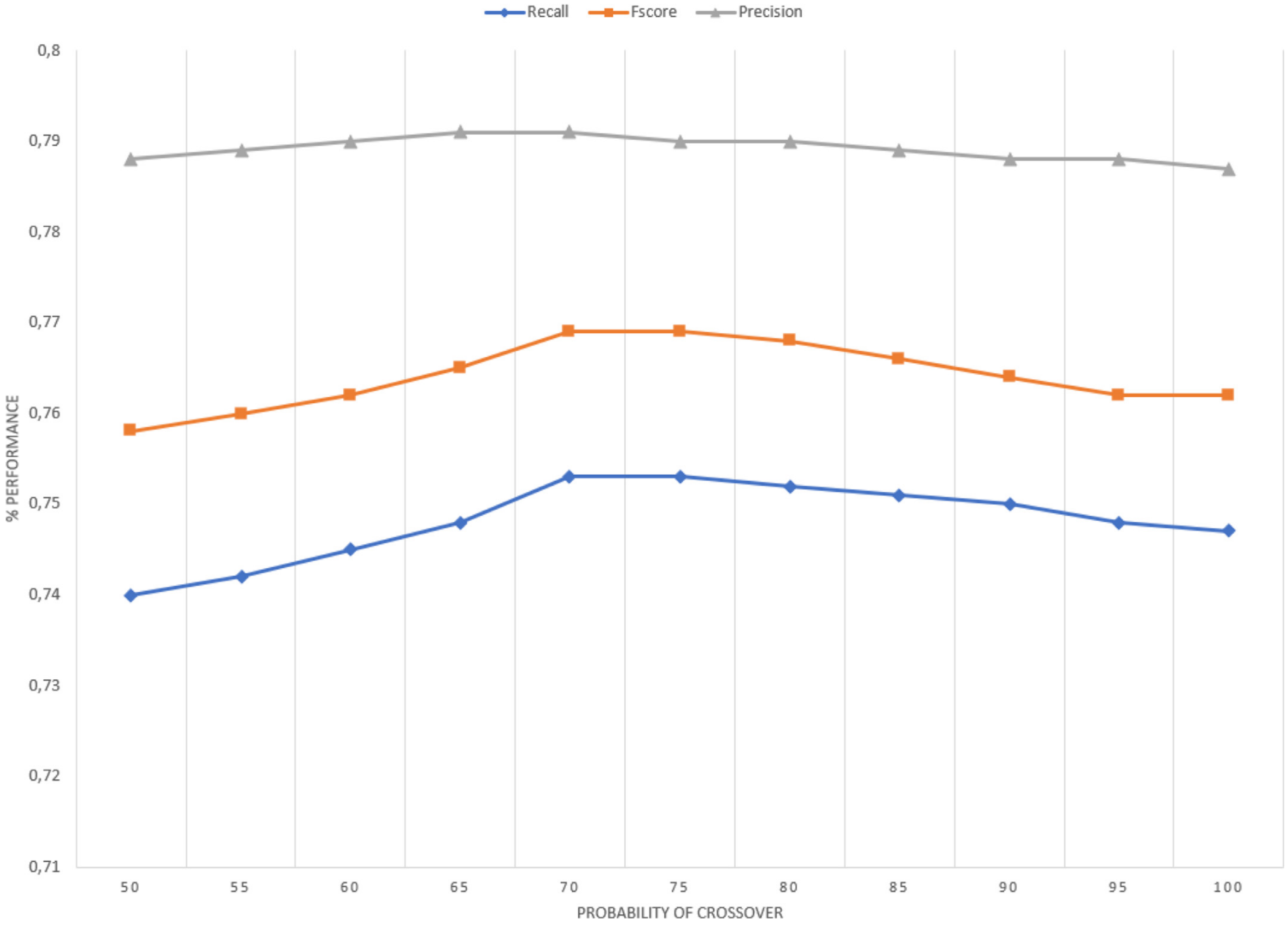


Fig. 12. Fitness evolution for the best solutions.

taining POS tags and dependency relations are created by using regular expressions and the *igraph* R package.

4.2. Data analysis

Performance evaluation of our GA-based model 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 Figs. 10 and 11 show the evolution of the fitness based on incremental values for P_c and P_m.

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 (Fig. 12).

Final coreference 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 (Fig. 13).

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 whenever some chains are not found on the annotated corpus, is not significant, suggesting the approach may achieve fair results without requiring a large annotated corpus.

Furthermore, comparisons with some SOTA approaches are 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 (Naredla & Adedoyin, 2022). Results show our approach ranks in a very competitive position for informal and short message CR.

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 (Sun, Liu, Chen, Hao, & Zhang, 2020). In addition, the majority of the approaches use formal and relatively large corpus so it is unclear whether they can perform well for different kind of texts.

5. Discussion

5.1. Contribution to literature

Our research proposes a novel approach using evolutionary computation techniques and natural-language processing methods in order to address that kind of underlying knowledge to improve further automated social media sentiment analysis.

Our method for sentiment analysis tasks outperformed other current approaches (Aktaş et al., 2020; Cai & Strube, 2010). A problem with pure classification task is the identification of subjective sentences according to the domain for the purpose of training the classifiers. henever a clas-

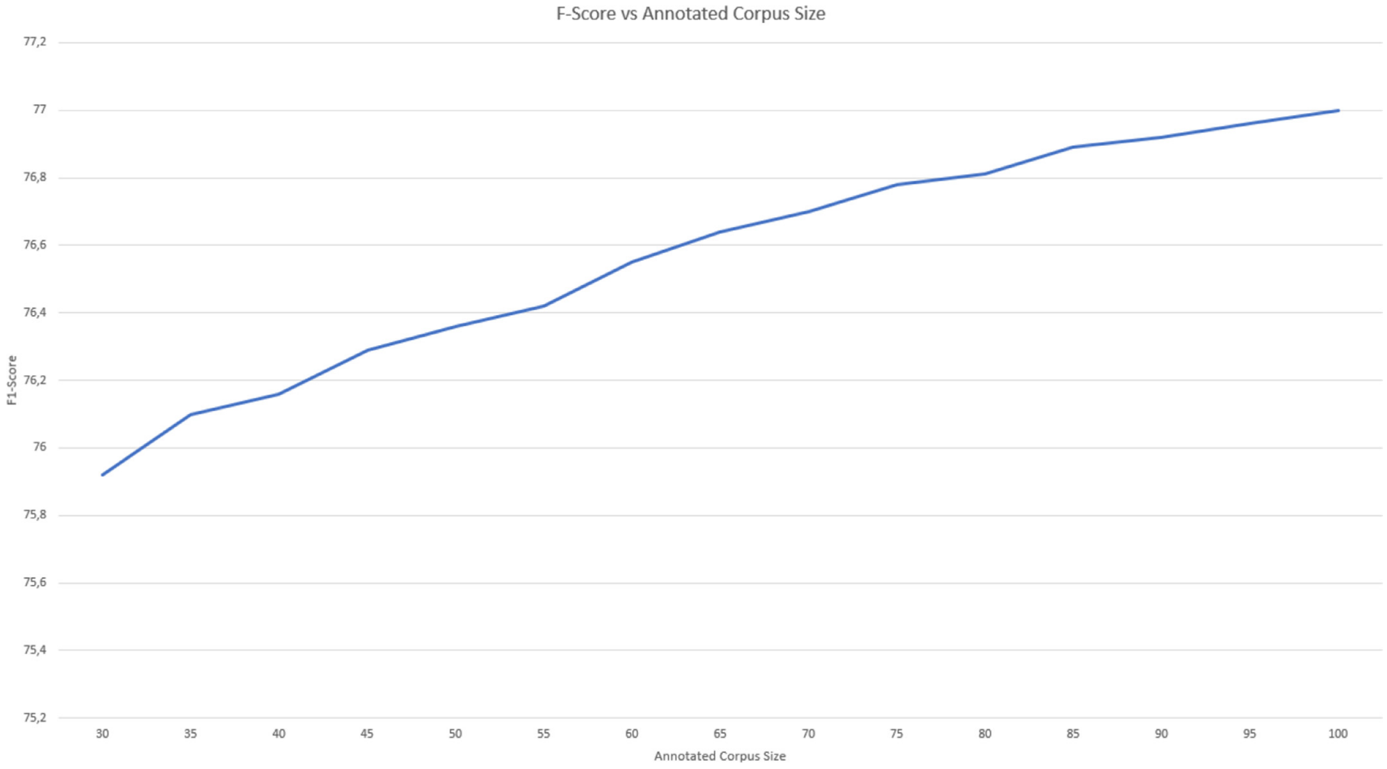


Fig. 13. F1-Score vs. annotated corpus size.

Table 1
Comparing our GA-based approach against SOTA.

Method	Type	Size of sample	Performance
Our GA-based approach	Informal short social media messages	Training involved 4525 tweets (24,090 words)	Precision = 0.80, Recall = 0.75 and $F1$ = 0.77
Naive Rule-based (baseline)	Formal computer manuals, nursing notes, Wall Street Journal articles	3900 sentences from the Treebank corpus	$F1 = 67.8\%$
Naive Classifier (Cai & Strube, 2010)	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 (Aktaş et al., 2020)	Informal short messages and news	MUC-7 (58.594 words)	Precision = 0.74, Recall = 62.45

Memory-based Learning (Atkinson et al., 2015)	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

sifier was trained on a specific domains, such classifiers would have low performance on other domains due to the lack of basic explicit features.

Unlike previously discussed approaches (Atkinson et al., 2015; Cai & Strube, 2010), our model becomes promising to combine linguistic coreference resolution methods and meta-heuristic optimization so that implicit objects and features can be identified before an information analysis task can be performed (Emami et al., 2019).

Furthermore, it is easy for our model's performance get improved by adding 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 analysis approaches.

5.2. *Practical implication*

This paper contributes an empirical study on coreference resolution methods for improving sentiment analysis tasks on social media. Our proposed model is helpful and effective for discovering implicit knowledge on comments, and messages on social media. If anyone wants to improve the coverage of messages retrieved for sentiment analysis applications, they can use our model.

6. Conclusions

In this work, a GA-based model for automatic coreference resolution for micro-blogging (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 SA.

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.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

John Atkinson: Conceptualization, Investigation. **Alex Escudero:** Conceptualization, Investigation.

References

Ahmad, S. N., & Laroche, M. (2017). Analyzing electronic word of mouth: A social commerce construct. *International Journal of Information Management, 37*(3), 202–213.

Aktaş, B., Solopova, V., Kohnert, A., & Stede, M. (2020). Adapting coreference resolution to twitter conversations. Association for computational linguistics: EMNLP 2020, November (p. 2454–2460).

Allam, H., Bliemel, M., Spiteri, L., Blustein, J., & Ali-Hassan, H. (2019). Applying a multi-dimensional hedonic concept of intrinsic motivation on social tagging tools: A theoretical model and empirical validation. *International Journal of Information Management, 45*, 211–222.

Atkinson, J. (2022). *Text analytics: An introduction to the science and applications of unstructured information analysis*. Chapman and Hall/CRC.

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

Batra, J., Jain, R., Tikkiwal, V. A., & Chakraborty, A. (2021). A comprehensive study of spam detection in e-mails using bio-inspired optimization techniques. *International Journal of Information Management Data Insights, 1*(1), 100006.

Bessou, S., & Aberkane, R. (2019). Subjective sentiment analysis for arabic newswire comments. *Journal of Digital Information Management.*

Cai, J., & Strube, M. (2010). Evaluation metrics for end-to-end coreference resolution systems. In *SIGDIAL '10: Proceedings of the 11th annual meeting of the special interest group on discourse and dialogue* (pp. 28–36).

Chen, R. R., Davison, R. M., & Ou, C. X. (2020). A symbolic interactionism perspective of using social media for personal and business communication. *International Journal of Information Management, 51*, 102022.

Darwich, M., Azman, S., Noah, M., Omar, N., & Osman, N. (2019). Corpus-based techniques for sentiment lexicon generation: A review. *Journal Of Digital Information Management, 17*, 289–296.

Derczynski, L., Maynard, D., Rizzo, G., van Erp, M., Gorrell, G., Troncy, R., … Bontcheva, K. (2015). Analysis of named entity recognition and linking for tweets. *Information Processing and Management, 51*(2), 32–49.

Emami, A., Trichelair, P., Trischler, A., & Suleman, K. (2019). The KnowRef coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution. In *ACL* (p. 9).

Fujii, M., Sakaji, H., Masuyama, S., & Sasaki, H. (2022). Extraction and classification of risk-related sentences from securities reports. *International Journal of Information Management Data Insights, 2*(2), 100096.

Georgiadou, E., Angelopoulos, S., & Drake, H. (2020). Big data analytics and international negotiations: Sentiment analysis of Brexit negotiating outcomes. *International Journal of Information Management, 51*, 102048.

Gupta, S., Kar, A. K., Baabdullah, A., & Al-Khowaiter, W. A. (2018). Big data with cognitive computing: A review for the future. *International Journal of Information Management, 42*, 78–89.

Hajli, N., & Featherman, M. S. (2017). Social commerce and new development in e–commerce technologies. *International Journal of Information Management, 37*(3), 177–178.

Hendrickx, I., & Hoste, V. (2009). Coreference resolution on blogs and commented newsanaphora processing and applications. In *7th Discourse Anaphora and Anaphor Resolution Colloquium, DAARC, Goa, India, November* (pp. 43–54).

Henkel, M., Perjons, E., & Sneiders, E. (2017). Examining the potential of language technologies in public organizations by means of a business and it architecture model. *International Journal of Information Management, 37*(1, Part A), 1507–1516.

Jabar, K., Alkhatib, B., & Wannous, H. (2022). A survey of sentiment analysis in the arabic language. *Journal of Digital Information Management.*

Kumar, S., Kar, A. K., & Ilavarasan, P. V. (2021). Applications of text mining in services management: A systematic literature review. *International Journal of Information Management Data Insights, 1*(1), 100008.

Lebib, F. Z., Mellah, H., & Drias, H. (2017). Enhancing information source selection using a genetic algorithm and social tagging. *International Journal of Information Management, 37*(6), 741–749.

Martarelli, N. J., & Nagano, M. S. (2022). How to undertake reviews of large collections of articles and establish main contributions: An ontology-based literature review approach. *International Journal of Information Management Data Insights, 2*(2), 100091.

Martin, N., Rice, J., & Arthur, D. (2020). Advancing social media derived information messaging and management: A multi-mode development perspective. *International Journal of Information Management, 51*, 102021.

Naredla, N. R., & Adedoyin, F. F. (2022). Detection of hyperpartisan news articles using natural language processing technique. *International Journal of Information Management Data Insights, 2*(1), 100064.

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights, 1*(1), 100007.

Noori, B. (2021). Classification of customer reviews using machine learning algorithms. *Applied Artificial Intelligence, 35*(8), 567–588.

Palivela, H. (2021). Optimization of paraphrase generation and identification using language models in natural language processing. *International Journal of Information Management Data Insights, 1*(2), 100025.

Shao, W., Hua, B., & Song, L. (2021). A pattern and pos auto-learning method for terminology extraction from scientific text. *Data and Information Management, 5*(3), 329–335.

Sharma, A., Rana, N. P., & Nunkoo, R. (2021a). Fifty years of information management research: A conceptual structure analysis using structural topic modeling. *International Journal of Information Management, 58*(C), 102316.

Sharma, S., Rana, V., & Kumar, V. (2021b). Deep learning based semantic personalized recommendation system. *International Journal of Information Management Data Insights, 1*(2), 100028.

Sheppard, C. (2016). *Genetic algorithms with Python*. CreateSpace Independent Publishing Platform.

Sukthanker, R., Poria, S., Cambria, E., & Thirunavukarasu, R. (2020). Anaphora and coreference resolution: A review. *Information Fusion, 59*, 139–162.

Sun, Y., Liu, X., Chen, G., Hao, Y., & Zhang, Z. J. (2020). How mood affects the stock market: Empirical evidence from microblogs. *Information and Management, 57*(5), 103181.

Wadud, A. H., Kabir, M. M., Mridha, M., Ali, M. A., Hamid, M. A., & Monowar, M. M. (2022). How can we manage offensive text in social media - a text classification approach using LSTM-boost. *International Journal of Information Management Data Insights, 2*(2), 100095.

Wang, Y., & Yu, C. (2017). Social interaction-based consumer decision-making model in social commerce: The role of word of mouth and observational learning. *International Journal of Information Management, 37*(3), 179–189.

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

Yerasani, S., Tripathi, S., Sarma, M., & Tiwari, M. K. (2020). Exploring the effect of dynamic seed activation in social networks. *International Journal of Information Management, 51*, 102039.

Yufang, H. (2020). Bridging anaphora resolution as question answering. In *Proceedings of the 58th annual meeting of the association for computational linguistics* (pp. 1428–1438). Association for Computational Linguistics.

Zhang, H., Song, Y., & Song, Y. (2019). Incorporating context and external knowledge for pronoun coreference resolution. CoRR abs/1905.10238

Zhang, J., Sun, Y., & Jara, A. J. (2015). Towards semantically linked multilingual corpus. *International Journal of Information Management, 35*(3), 387–395.