Engineering Applications of Artificial Intelligence 46 (2015) 279–286

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Combining Singular Value Decomposition and a multi-classifier: A new approach to support coreference resolution

Ana Zelaia *, Olatz Arregi, Basilio Sierra

Faculty of Informatics, University of the Basque Country, UPV/EHU, Manuel Lardizabal Pasealekua 1, 20018 Donostia-San Sebastián, Basque Country, Spain

ARTICLE INFO

Article history:
Received 20 May 2015
Received in revised form
11 September 2015
Accepted 16 September 2015
Available online 23 October 2015

Keywords:
Coreference resolution
Machine learning
Multi-classifier
Singular Value Decomposition
Latent semantic indexing

ABSTRACT

In this paper a new machine learning approach is presented to deal with the coreference resolution task. This approach consists of a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space. The vector representation for mention-pairs is generated using a rich set of linguistic features. The (Singular Value Decomposition) SVD technique is used to generate the reduced dimensional vector space. The approach is applied to the OntoNotes v4.0 Release Corpus for the column-format files used in CONLL-2011 coreference resolution shared task. The results obtained show that the reduced dimensional representation obtained by SVD is very adequate to appropriately classify mention-pair vectors. Moreover, it can be stated that the multi-classifier plays an important role in improving the results.

1. Introduction

Coreference resolution deals with the problem of finding all expressions that refer to the same entity in a text (Mitkov, 2002). It is an important subtask in Natural Language Processing (NLP) tasks that require natural language understanding, and hence, it is considered to be difficult.

A coreference resolution system has to automatically identify the mentions of entities in text and link the corefering mentions (the ones that refer to the same entity) to form coreference chains. Systems are expected to perform both, mention detection and coreference resolution.

Preliminary researches proposed heuristic approaches to the task, but thanks to the annotated coreference corpora made available in the last years and the progress achieved in statistical NLP methods, machine learning approaches to the coreference resolution task are being proposed. In Ng (2010) the authors present an interesting survey of the progress in coreference resolution.

In this paper a new machine learning approach is presented to deal with the coreference resolution task. Given a corpus with annotated mentions, the multi-classifier system presented classifies mention-pairs in a reduced dimensional vector space. The typical mention-pair model is used, where each pair of mentions is represented by a rich set of linguistic features; positive instances correspond to mention-pairs that corefer. In this paper, coreference resolution is tackled as a binary classification problem (Soon et al., 2001); the subsequent linking of mentions into coreference chains is not considered. In fact, the aim of the experiment performed is to measure to what extent working with feature vectors in a reduced dimensional vector space and applying a multi-classifier system helps to determine the coreference of mention-pairs. To the best of our knowledge, there are no approaches to the coreference resolution task which make use of multi-classifier systems to classify mention-pairs in a reduced dimensional vector space.

This paper gives a description of a new approach to deal with the problem of identifying whether two mentions corefer and shows the results obtained. Section 2 presents related work. In Section 3 the new approach is presented. Section 4 presents the case study, where details about the dataset used in the experiments and the preprocessing applied are given. In Section 5 the experimental setup is presented. The experimental results are shown and discussed in Section 6, and finally, Section 7 contains some conclusions and comments on future work.

2. Related work

Much attention has been paid to the problem of coreference resolution in the past two decades. Conferences specifically focusing coreference resolution have been organized since 1995. The sixth

* Corresponding author.
E-mail addresses: ana.zelaia@ehu.eus (A. Zelaia), olatz.arregi@ehu.eus (O. Arregi), b.sierra@ehu.eus (B. Sierra).

http://dx.doi.org/10.1016/j.engappai.2015.09.007
0952-1976/

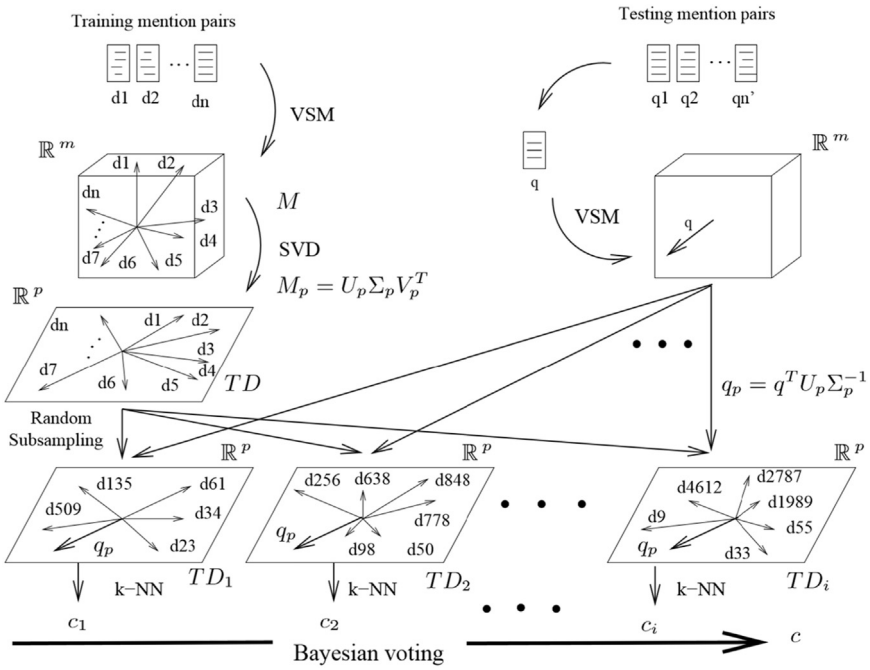

Fig. 1. Fundamental steps of the proposed approach. Original representation for mention-pairs: $\mathbb{R}^m$. SVD-dimensional vector representation computed by LSI: $\mathbb{R}^p$. Multi-classifier constructed based on training datasets $TD_1, \ldots, TD_i$ and several k-NN classifiers. Testing mention-pair q is projected to the SVD-dimensional vector space. Label predictions are combined to compute the final c: (+) mentions corefer, (−) they do not corefer.

and seventh Message Understanding Conferences included a specific task on coreference resolution (MUC6, 1995; Hirschman and Chinchor, 1998). The Automatic Context Extraction (ACE) Program focused on identifying certain types of relations between a predefined set of entities (Doddington et al., 2004) while the Anaphora Resolution Exercise (ARE) involved anaphora resolution and Noun Phrase coreference resolution (Orasan et al., 2008).

More recently, SemEval-2010 Task 1 was dedicated to coreference resolution in multiple languages. One year later, in the CoNLL-2011 shared task (Pradhan et al., 2011), participants had to model unrestricted coreference in the English-language OntoNotes corpora and CoNLL-2012 Shared Task (Pradhan et al., 2012) involved predicting coreference in three languages: English, Chinese and Arabic.

Recent work on coreference resolution has been largely dominated by machine learning approaches. In the SemEval-2010 task on Coreference Resolution in Multiple Languages (Recasens et al., 2010), most of the systems were based on these techniques (Broscheit et al., 2010; Uryupina, 2010; Kobdani and Schütze, 2010). The same occurred at CoNLL-2011, where Chang et al. (2011), Björkelund and Nugues (2011), and Nogueira dos Santos and Lopes Carvalho (2011) were based on machine learning techniques. There are many open-source platforms and machine learning based coreference systems such as BART (Versley et al., 2008) and the Illinois Coreference Package (Bengtson and Roth, 2008), among others.

Nevertheless, rule-based systems have also been applied successfully (Lappin and Leass, 1994; Mitkov, 1998; Lee et al., 2013). The authors of Lee et al. (2013) propose a coreference resolution system that is an incremental extension of the multi-pass sieve system proposed in Raghunathan et al. (2010). This system is shifting from the supervised learning setting to an unsupervised setting, and obtained the best result in the CoNLL-2011 Shared Task. It is integrated in the Stanford CoreNLP toolkit (Manning et al., 2014).

Some very interesting uses of vector space models for the coreference resolution task can be found in the literature. In Nilsson and Hjelm (2009) the authors investigate the effect of using vector space models as an approximation of the kind of lexico-semantic and common-sense knowledge needed for coreference resolution for Swedish texts. They also work with reduced dimensional vector spaces and obtain encouraging results. In an attempt to increase the performance of a coreference resolution engine, structured semantic knowledge available in the web is used in Bryl et al. (2010). One of the strategies they adopt is to apply the SVD to Wikipedia articles and classify mentions in a reduced dimensional vector space.

3. Proposed approach

The approach presented in this paper consists of a multi-classifier system which classifies mention-pairs in a reduced dimensional vector space. This multi-classifier is composed of several k-Nearest Neighbors (k-NN) classifiers. A set of linguistic features is used to generate the vector representations for the mention-pairs. The training dataset is used to create a reduced dimensional vector space using the SVD technique. Mention-pairs in the training, development and testing sets are represented using the same linguistic features and projected onto the reduced dimensional space.

The classification process is performed in the reduced dimensional space. To create the multi-classifier, random subsampling is applied and $TD_1, \ldots, TD_i$ training datasets are obtained for the reduced dimensional space. Given a testing case q, the k-NN classifiers make label predictions $c_1, \ldots, c_i$ based on the training datasets $TD_1, \ldots, TD_i$. These predictions are combined to obtain the final prediction c using a Bayesian voting scheme and based on the confidence values computed. It is a binary classification system where the final prediction c may be positive (mentions tested corefer) or negative (mentions do not corefer). Fig. 1 shows the fundamental steps of the experiment.

In the rest of this section, details about the SVD dimensionality reduction technique, the k-NN classification algorithm, the combination of classifiers and the evaluation measures used are briefly reviewed.

3.1. The SVD dimensionality reduction technique

The classical Vector Space Model (VSM) has been successfully employed to represent documents in text categorization and Information Retrieval tasks. Latent Semantic Indexing (LSI)[1] is a variant of the VSM in which documents are represented in a lower dimensional vector space created from a training dataset (Deerwester et al., 1990). To create such a lower dimensional vector space, LSI generates a term-document matrix M and computes its Singular Value Decomposition (SVD) matrix decomposition, $M = U\Sigma V^T$. As a result, r singular values are obtained, and terms and documents are mapped to the r-dimensional vector space. By reducing the r to p, a reduced dimensional space is created, the p-dimensional space onto which vectors are projected. This reduced dimensional space is used for classification purposes, and the cosine similarity is usually used to measure the similarity between vectors (Berry et al., 1995).

It has been proved that computing the similarity of vectors in the reduced dimensional space gives better results than working in the original space. In fact, LSI is said to be able to capture the latent relationships among words in documents thanks to the word co-occurrence analysis performed by the SVD technique, and therefore, cluster semantically terms and documents. This powerful technique is being used to better capture the semantics of texts in applications such as Information Retrieval (Berry and Browne, 2005). LSI is referred to as Latent Semantic Analysis (LSA) when it is used as a model of the acquisition, induction and representation of language and the focus is on the analysis of texts (Dumais, 2004).

For the sake of the coreference resolution task, each document corresponds to a mention-pair, and words in each document are the linguistic feature values for the associated mention-pair. Matrix M is constructed for the selected feature values (terms) and all mention-pairs considered (documents) in the training dataset. The SVD decomposition is computed and the p-dimensional reduced space is created. In the approach presented U is used as the reduced dimensional representation, and the coordinates are computed to project mention-pair vectors onto the reduced space and compare them.

3.2. The k-NN classification algorithm

The k-Nearest Neighbors algorithm (k-NN) is a distance based classification approach. According to this approach, given an arbitrary testing case, the k-NN classifier ranks its nearest neighbors among the training cases, and uses the class of the k top-ranking neighbors to do the prediction for the testing case being analyzed (Dasarathy, 1991; Aha et al., 1991).

Parameter k is set to 3 in the approach presented, based on our previous experiments (Zelaia et al., 2005). Given a testing mention-pair vector q, the 3-NN classifier is used to find the three nearest neighbor mention-pair vectors in the reduced dimensional vector space. The cosine is used to measure vector similarity and find the nearest.

In this paper, the k-NN classifier provided with the Weka package (Hall et al., 2009) is also used. Results obtained with it are considered a baseline and make it possible to provide a honest comparison to the ones obtained with the proposed approach.

3.3. Multi-classifier systems

The combination of multiple classifiers has been intensively studied with the aim of improving the accuracy of individual components (Ho et al., 1994). A widely used technique to implement this approach is *bagging* (Breiman, 1996), where a set of training datasets TD_i is generated by selecting n training cases drawn randomly with replacement from the original training dataset TD of n cases. When a set of $n_1 < n$ training cases is chosen from the original training collection, the bagging is said to be applied by random subsampling. In the approach presented in this paper, parameter n_1 is set to be 60% of the total number of training cases n, based on some previous experiments carried out for this task. The proportion of positive and negative cases in the training dataset TD is preserved in the different TD_i datasets generated.

Given a testing case q, the multi-classifier makes label predictions $c_1, \ldots, c_i$ based on each one of the training datasets $TD_1, \ldots, TD_i$. These label predictions may be either positive (+) or negative (−). One way to combine the predictions is by Bayesian voting (Dietterich, 1998), where a confidence value is calculated for each training dataset TD_j, $j = 1, \ldots, i$ and label to be predicted ($c = +, c = -$): $cv^j_{(+)}$, $cv^j_{(-)}$. These confidence values are calculated based on the training collection. Confidence values are summed by label; the label c that gets the highest value is finally proposed as a prediction for the testing case q.

3.4. Evaluation measures

The approach presented in this paper is a binary classification system where the final prediction c may be positive (mentions tested corefer) or negative (mentions do not corefer). There are many metrics that can be used to measure the performance of a classifier. In binary classification problems precision and recall are very widely used. Precision (Prec) is the number of correct positive results divided by the number of all positive results, and recall (Rec) is the number of correct positive results divided by the number of positive results that should have been returned.

In general, there is a trade-off between precision and recall. Thus, a classifier is usually evaluated by means of a measure which combines them. The F_1-score can be interpreted as a weighted average of precision and recall:

$$F_1 = \frac{2 \cdot \mathrm{Prec} \cdot \mathrm{Rec}}{\mathrm{Prec} + \mathrm{Rec}}$$

Accuracy is also used as a statistical measure of performance in binary classification tasks. Accuracy is the proportion of true results (both true positives and true negatives) among the total number of cases tested.

4. Case study

This section briefly reviews the dataset used in the experiments and the preprocessing applied.

4.1. Dataset

The OntoNotes v4.0 Release Corpus is used in the experiments.[2] It provides a large-scale multi-genre corpus with multiple layers of annotation (syntactic, semantic and discourse information) which also include coreference tags. A nice description of the coreference annotation in OntoNotes can be found in Pradhan et al. (2007a) and Pradhan et al. (2007b).

Although OntoNotes is a multi-lingual resource for English, Chinese and Arabic, for the scope of this paper, just the English texts for five different genres or types of sources are used:

[1] http://lsi.research.telcordia.com,http://www.cs.utk.edu/~lsi.

[2] Downloaded from Linguistic Data Consortium (LDC) Catalog No.: LDC2011T03, https://catalog.ldc.upenn.edu/LDC2011T03. For more information, see OntoNotesRelease4.0.pdf and coreference/englishcoref.pdf files in LDC directory.

```
#begin document (bn/abc/00/abc_0011); part 000
bn/abc/00/abc_0011 0  0 President NNP  (TOP(S(NP* - - - -   * (ARG1* (ARG0*   (0
bn/abc/00/abc_0011 0  1 Clinton   NNP  *)         - - - -   (PERSON) *) *)     0)
bn/abc/00/abc_0011 0  2 is        VBZ  (VP*       be 01 1 - * (V*)     *     -
bn/abc/00/abc_0011 0  3 on        IN   (PP*       - - - -   * (ARG2*   *     -
bn/abc/00/abc_0011 0  4 his       PRP (NP(NP*    - - - -   * *        *      (0)
bn/abc/00/abc_0011 0  5 way       NN    *)        - - 4 -   * *        *     -
bn/abc/00/abc_0011 0  6 to        IN   (PP*       - - - -   * *        *     -
bn/abc/00/abc_0011 0  7 Egypt     NNP  (NP*))))   - - - -   (GPE)  *)  *     -
bn/abc/00/abc_0011 0  8 to        TO   (S(VP*     - - - -   * (ARGM-PRP*  *  -
bn/abc/00/abc_0011 0  9 attend    VB   (VP*       attend 01 1 - * * (V*)     -
bn/abc/00/abc_0011 0 10 tomorrow  NN   (NP(NP*    - - - -   (DATE) * (ARG1*   (1
bn/abc/00/abc_0011 0 11 's        POS   *)        - - - -   * *        *       1)
bn/abc/00/abc_0011 0 12 emergency NN    *         - - 1 -   * *        *     -
bn/abc/00/abc_0011 0 13 summit    NN    *)))))    - - 3 -   * *)       *)    -
bn/abc/00/abc_0011 0 14 .         .     *))       - - - -   * *        *     -
bn/abc/00/abc_0011 0  0 Tensions  NNS   (TOP(S(S(NP(NP*)- - - - * (ARG1* * * -
bn/abc/00/abc_0011 0  1 in        IN    (PP*      - - - -   * * *      *     -
bn/abc/00/abc_0011 0  2 the       DT    (NP*      - - - -  (LOC* * *   *     -
bn/abc/00/abc_0011 0  3 Middle    NNP    *        - - - -   *    * *   *     -
bn/abc/00/abc_0011 0  4 East      NNP    *)))     - - - -   *) *)  *   *     -
bn/abc/00/abc_0011 0  5 remain    VBP   (VP*      remain 01 1 - * (V*) * *   -
bn/abc/00/abc_0011 0  6 very      RB    (ADJP*    - - - -   * (ARG3* * *     -
bn/abc/00/abc_0011 0  7 high      JJ     *)       - - - -   * *)     * *     -
bn/abc/00/abc_0011 0  8 after     IN    (PP*      - - - -   * (ARGM-TMP* * * -
bn/abc/00/abc_0011 0  9 two       CD    (NP(NP*   - - - -   (DATE* * * *     -
bn/abc/00/abc_0011 0 10 weeks     NNS    *)       - - - -   *) *   * *       -
bn/abc/00/abc_0011 0 11 of        IN    (PP*      - - - -   *  *   * *       -
bn/abc/00/abc_0011 0 12 violence  NN    (NP*))))))- - - -   *  *)  * *       -
bn/abc/00/abc_0011 0 13 and       CC     *        - - - -   *  *   * *       -
bn/abc/00/abc_0011 0 14 the       DT    (S(NP(NP* - - - -   *  * (ARG1* *    -
bn/abc/00/abc_0011 0 15 immediate JJ     *        - - - -   *  *   *    *    -
bn/abc/00/abc_0011 0 16 goal      NN     *)       - - 1 -   *  *   *    *    -
bn/abc/00/abc_0011 0 17 tomorrow  NN    (NP*))    - - - -  (DATE) * *)  *     (1)
bn/abc/00/abc_0011 0 18 is        VBZ   (VP*      be 01 1 - *  * (V*)   *    -
bn/abc/00/abc_0011 0 19 to        TO    (S(VP*    - - - -   *  * (ARG2* *    -
bn/abc/00/abc_0011 0 20 stop      VB    (VP*      stop 01 2 - * * *  (V*)    -
bn/abc/00/abc_0011 0 21 the       DT    (NP*      - - - -   * * * (ARG1*     -
bn/abc/00/abc_0011 0 22 killing   NN     *))))))  - - 1 -   * * *)     *)    -
bn/abc/00/abc_0011 0 23 .          .     *))      - - - -   * * *      *     -
#end document
```

Fig. 2. An example of *_conll file. There are four coreference mentions: m_1=*President Clinton*, m_2=*his*, m_3=*tomorrow's*, m_4=*tomorrow* and two coreference chains: {*President Clinton*, *his*} and {*tomorrow's*, *tomorrow*}.

broadcast conversations (BC), broadcast news (BN), magazine articles (MZ), newswires (NW) and web data (WB).

The English language portion of the OntoNotes v4.0 Release Corpus was used in the CONLL-2011 coreference resolution Shared task.[3] The task is to automatically identify mentions of entities and events in text and to link the corefering mentions together to form mention chains (Pradhan et al., 2011, 2012). Since OntoNotes coreference data spans multiple genre, the task organizers created a testing set spanning all the genres. The training, development and testing files are downloaded from the CONLL-2011 website. In this work, hand-annotated gold files are used for the experiments. The *_conll files contain information in a tabular structure where the last column contains coreference chain information. The example of Fig. 2 shows a *_conll file with four coreference mentions annotated: Mentions m_1=*President Clinton* and m_2=*his* are coreferent and therefore have the same label (0) in the last column, and mentions m_3=*tomorrow's* and m_4=*tomorrow*, which are also coreferent, have label (1). These four mentions form two coreference chains: {*President Clinton*, *his*} and {*tomorrow's*, *tomorrow*}.

4.2. Preprocessing

In order to obtain the vector representation for each pair of mentions, the features defined in Sapena et al. (2011) and Sapena et al. (2013) are used. The authors of the cited papers developed a coreference resolution system called RelaxCor[4] and participated in

[3] http://conll.cemantix.org/2011/introduction.html.

[4] http://nlp.lsi.upc.edu/relaxcor/.

the CoNLL-2011 shared task obtaining very good results. It is an open source software available for anyone who wishes to use it. Results computed for these original feature vectors are used as a baseline for the proposed approach.

The 127 binary features used contain morphosyntactic and lexicosemantic information. These features are related to the distance between the two mentions (in the same sentence, in consecutive sentences, is the first mention, etc.), lexical information (string matching of mentions, both are pronouns and their strings match, etc.), morphological information (the number of both mentions matches, the gender of both mentions matches, etc.), syntactic dependencies (one mention is included in the other, etc.) and semantic information (the same semantic role, one mention is an alias of the other, etc.). Using the coreference information given in the *_conll files, mention-pairs are generated, their corresponding feature vector is created and a label is assigned to each of them: a positive label (+) indicates that the two mentions corefer, whereas a negative label (−) indicates that they do not corefer. According to the example of Fig. 2, there are two positive mention-pairs: m_1–m_2 and m_3–m_4. There are four more possible mention-pairs, all of which are negative: m_1–m_3, m_1–m_4, m_2–m_3 and m_2–m_4.

Note that each mention in a file is combined with all the rest of mentions in the same file to form mention-pairs. Pairing the four mentions in Fig. 2, for example, six mention-pairs can be generated, only two of which are positive. Consequently, a very large amount of negative instances is generated, specially for large files. In order to reduce the amount of negative instances in a similar manner as in Sapena et al. (2011), negative instances with more than five feature values different from any positive instance in each file are eliminated. Bringing together the instances generated from files of the same split and genre, the training, development and testing corpora for the 5 genres are created. Contradictions (negative instances with identical feature values as a positive instance) and instances that appear more than once in the same corpus are removed. Since the size of the corpora generated was too large for some of the genres, a stratified random sampling strategy is applied to reduce the size of all corpora; the broadcast conversations (BC) genre training corpus, for example, had more than 4 million instances before the size reduction strategy was applied. Table 1. gives detailed information about the number of positive and negative mention-pairs in the training, development and testing corpora used in the experiments.

Applying Latent Semantic Indexing (LSI) a term-document matrix is constructed for each of the five training corpora. Documents represent mention-pairs and each of them consists of 127 words (linguistic feature values) out of the 254 possible ones. The ones selected by LSI are assigned a row in the term-document matrix. Feature values found in each corpus are indexed and counted in order to compute a table of documents and words. Only feature values that appear above an established frequency threshold in the training corpora are selected as terms. A matrix that reflects whether each term appears in each document is created. Note that this matrix is binary.

Table 2 shows the number of terms (selected feature values) and documents (positive and negative mention-pairs) found in the training corpora for each genre. The third row in the table shows the number of singular values (dimensions) computed by SVD for each of the genres. These values are quite similar for the five genres, ranging from 83 to 87. By means of these SVD-dimensions the SVD-dimensional vector representation for the documents (mention-pairs) is obtained.

Table 1
Size of corpora used in the experiments.

	BC	BN	MZ	NW	WB
Training (+)	20,206	44,515	25,103	31,034	24,501
Training (−)	26,623	55,921	23,568	50,687	26,948
Development (+)	4,056	5,920	3,873	4,776	3,531
Development (−)	5,831	8,609	4,864	7,615	5,732
Testing (+)	29,363	10,771	3,918	15,857	17,146
Testing (−)	16,591	12,480	3,209	15,759	5,505

Table 2
Terms, documents and singular values (SVD-dimensions) for the five training corpora.

	BC	BN	MZ	NW	WB
Terms (selected feature values)	227	230	227	229	230
Documents (mention-pairs)	46,829	100,436	48,671	81,721	51,449
Singular Values (SVD-dimensions)	83	86	85	86	87

5. Experimental setup

To optimize the behavior of the proposed approach, the five development corpora are used to adjust two parameters in a parameter tuning phase. The two parameters optimized are the dimension of the vector space and the number of classifiers for the multi-classifier system.

- *The dimension of the vector space*: the reduction of the SVD-dimension is analyzed to see if results improve by means of a reduced dimensional representation for mention-pairs. The following values are experimented: 5, 10, 15, 20, 25, 30, 40.
- *The number of classifiers*: To optimize the behavior of the multi-classifier system, the number of training datasets is adjusted. The following values are experimented: 5, 10, 20, 30, 40, 50, 60, 70, 80.

The five genres correspond to texts coming from different sources and may have very different characteristics (Uryupina and Poesio, 2012). That is why they are treated as five different classification problems and therefore, the parameter optimization process is performed in an independent way for each of the genres.

Tables 3–7 show the results for the different values of the two parameters using the development corpora. Rows in the tables correspond to values for the reduced dimension, and columns correspond to the number of classifiers. Two evaluation measures are computed for each parameter-pair; results in the first row are accuracy rates; the ones in the second, F_1-scores. The highest value in each row is shown in bold, and the highest F_1-score in each table is shown in a box.

The optimal values for parameters are determined by the highest F_1-score in each table (the values in a box) and are summarized in Table 8. According to these optimal values, testing mention-pair vectors for the BC genre, for example, are projected onto the 30-dimensional vector space and classified by a multi-classifier formed by 60 k-NN classifiers. This implies that 60 training datasets (TD_i) have to be sampled in the reduced 30 dimensional space and each one is used to obtain a classification for a given testing mention-pair using the k-NN classifier.

6. Experimental results

In order to evaluate the impact of LSI in this task, some experiments are carried out in the testing phase.

- *Baseline*: To compute a baseline for the proposed approach, the classification of testing mention-pairs represented by the

Table 3
Parameter tuning for the BC genre. Accuracy and F_1-score.

BC genre		Number of TD_i training datasets							
Dimension		5	10	20	30	40	50	60	70
10	Acc.	67.40	**67.99**	67.68	67.86	67.69	67.80	67.96	67.87
	F_1	57.99	57.75	57.88	**58.42**	58.18	58.39	58.37	58.41
15	Acc.	65.26	**66.61**	66.19	66.43	66.22	66.30	66.38	66.39
	F_1	58.91	58.79	59.39	59.43	59.44	59.66	59.64	**59.73**
20	Acc.	65.51	66.67	**66.77**	66.50	66.75	66.07	66.02	66.20
	F_1	59.97	60.23	60.67	60.33	**60.75**	60.30	60.18	60.22
25	Acc.	64.86	65.85	66.06	66.15	66.17	66.16	**66.22**	66.15
	F_1	60.15	60.12	60.40	60.63	60.85	60.78	60.84	**60.89**
30	Acc.	64.68	66.25	65.93	66.13	65.85	65.98	**66.44**	66.18
	F_1	60.60	61.10	61.49	61.49	61.50	61.66	**61.85**	61.71
40	Acc.	65.75	64.50	65.52	**66.11**	65.69	65.82	65.65	65.70
	F_1	61.31	61.06	61.16	61.40	61.17	**61.47**	61.17	61.24

Table 4
Parameter tuning for the BN genre. Accuracy and F_1-score.

BN genre		Number of TD_i training datasets					
Dimension		5	10	20	30	40	50
10	Acc.	70.13	70.90	**71.42**	71.30	70.96	71.12
	F_1	65.42	65.25	66.08	**66.32**	66.07	66.05
15	Acc.	68.08	**69.75**	69.59	69.65	69.50	69.44
	F_1	64.01	64.32	64.61	**64.88**	64.66	64.82
20	Acc.	70.55	**72.15**	71.85	71.84	71.69	71.82
	F_1	66.14	66.63	66.85	67.01	66.96	**67.17**
25	Acc.	70.70	71.91	71.84	**72.16**	71.98	71.98
	F_1	65.97	66.33	66.68	**67.36**	67.09	67.21
30	Acc.	69.99	**71.98**	71.93	71.91	71.80	71.96
	F_1	65.40	66.25	66.69	66.92	66.75	**66.97**
40	Acc.	70.58	72.13	72.09	71.81	**72.20**	71.99
	F_1	65.56	66.21	66.51	66.33	**66.95**	66.68

Table 5
Parameter tuning for the MZ genre. Accuracy and F_1-score.

MZ genre		Number of TD_i training datasets						
Dimension		5	10	20	30	40	50	60
10	Acc.	63.84	63.29	63.33	64.40	64.11	64.36	**64.53**
	F_1	65.59	65.96	65.66	66.34	66.11	**66.36**	66.30
15	Acc.	65.29	65.45	65.92	66.25	**66.26**	66.17	66.20
	F_1	66.52	67.13	67.31	**67.43**	67.41	67.30	67.24
20	Acc.	64.71	65.27	66.05	65.93	65.93	**66.41**	66.00
	F_1	66.57	67.34	67.87	67.73	67.63	**68.10**	67.66
25	Acc.	64.94	65.10	65.03	65.99	65.68	65.86	**66.06**
	F_1	66.86	67.53	67.21	67.78	67.56	67.57	**67.81**
30	Acc.	65.19	64.47	64.96	65.07	65.35	**65.51**	65.51
	F_1	67.02	67.11	67.31	67.31	67.30	**67.53**	67.48
40	Acc.	65.22	64.89	64.65	65.25	**65.46**	65.37	65.21
	F_1	67.34	67.42	66.98	**67.52**	**67.52**	67.36	67.36

original 127 binary features is considered. They are also used by RelaxCor, the existing most similar method to the proposed approach. Mention-pairs are classified using a single 3-NN classifier.

- *Single classification*: In a second experiment, some very widely used standard classification algorithms such as Naive Bayes (NB), classification trees (C4.5), Support Vector Machines (SVM) and *k*-nearest neighbors are used to classify mention-pairs represented in the SVD-dimensional vector space created by LSI (see the SVD-dimensions used for the five genres in Table 2).
- *Proposed approach*: In a third experiment the proposed approach is used. First, a multi-classifier system composed of several 3-NN classifiers classifies testing mention-pairs in the same SVD-dimensional vector space as in the previous experiment (MultiCl.$_{opt}$+ SVD). This multi-classifier is generated according to the optimal number of classifiers for each genre (see Table 8). Finally, the same multi-classifier is applied for the optimal SVD-dimensions per genre (MultiCl.$_{opt}$+SVD$_{opt}$) (see optimal number of classifiers and optimal SVD-dimensions in Table 8).

Table 6
Parameter tuning for the NW genre. Accuracy and F_1-score.

NW genre		Number of TD_i training datasets				
Dimension		5	10	20	30	40
10	Acc.	77.64	77.94	**78.39**	78.05	78.33
	F_1	69.58	69.17	69.85	69.62	**69.99**
15	Acc.	77.12	78.15	**78.44**	78.34	78.21
	F_1	69.21	69.62	**70.19**	**70.19**	70.18
20	Acc.	76.95	78.37	78.15	78.29	**78.38**
	F_1	69.27	70.01	69.87	70.20	**70.41**
25	Acc.	77.15	78.41	**78.73**	78.54	78.52
	F_1	69.48	70.02	**70.85**	70.73	70.69
30	Acc.	76.41	77.99	77.75	**78.16**	78.07
	F_1	68.44	69.09	69.24	**69.91**	69.82
40	Acc.	77.33	**78.58**	78.35	78.42	78.52
	F_1	69.80	70.26	70.29	70.56	**70.60**

Table 7
Parameter tuning for the WB genre. Accuracy and F_1-score.

WB genre		Number of TD_i training datasets					
Dimension		5	10	20	30	40	50
5	Acc.	66.53	67.73	67.77	**67.90**	67.64	67.46
	F_1	62.02	62.48	62.41	**62.64**	62.47	62.32
10	Acc.	67.51	**67.91**	67.62	67.75	67.79	67.47
	F_1	65.22	65.05	65.03	65.45	**65.49**	65.06
15	Acc.	65.31	**67.61**	67.13	66.96	67.11	67.30
	F_1	62.87	64.14	64.11	64.05	64.36	**64.50**
20	Acc.	66.03	**67.39**	67.24	66.91	66.92	67.12
	F_1	63.18	63.73	**64.27**	64.00	64.03	**64.27**
25	Acc.	65.21	66.65	**66.69**	66.51	66.63	66.37
	F_1	62.70	63.21	63.71	63.57	**63.89**	63.66
30	Acc.	64.92	**66.55**	65.88	66.04	66.14	65.86
	F_1	62.33	62.80	62.88	63.04	**63.25**	63.09
40	Acc.	65.25	**66.54**	66.30	66.06	66.13	66.32
	F_1	62.94	63.45	63.43	63.45	63.78	**63.92**

Table 8
Optimal dimension and number of classifiers.

Optimal parameters	BC	BN	MZ	NW	WB
Optimal dimension	30	25	20	25	10
Optimal number of classifiers	60	30	50	20	40

Table 9 shows the results obtained in each of the experiments. The results shown in bold in the columns that correspond to the five genres are the best accuracy and F_1-score for each genre. Note

Table 9
Testing results for the five genres. Last column: mean accuracy and F_1-scores.

Experiment		BC	BN	MZ	NW	WB	Mean
Baseline	Acc.	**71.90**	70.40	**70.60**	70.70	66.90	70.10
(RelaxCor)	F_1	**76.20**	68.60	**73.10**	67.90	74.40	72.00
Single classification	Acc.	42.21	61.90	61.95	66.76	41.85	54.93
(NB + SVD)	F_1	34.90	35.90	58.30	57.00	43.00	45.82
Single classification	Acc.	64.86	69.76	65.32	70.82	69.67	68.09
(C4.5 + SVD)	F_1	72.70	64.70	71.10	69.70	78.10	71.26
Single classification	Acc.	63.62	51.98	68.75	68.62	70.92	64.78
(SVM + SVD)	F_1	68.60	3.80	71.10	59.90	79.70	56.62
Single classification	Acc.	67.20	72.50	66.20	72.50	**78.30**	71.30
(3-NN + SVD)	F_1	74.20	71.00	71.70	71.50	**85.00**	**74.70**
Proposed approach	Acc.	66.90	**75.50**	66.10	**74.20**	77.60	**72.10**
MultiCl.$_{opt}$ + SVD	F_1	73.90	**72.80**	70.70	**71.60**	84.10	74.60
Proposed approach	Acc.	66.30	74.30	68.50	71.20	76.20	71.30
MultiCl.$_{opt}$ + SVD$_{opt}$	F_1	72.40	71.40	71.50	67.80	83.10	73.20

that the two performance measures computed are very correlated in the five cases. Taking into account that the proportion of positive and negative instances varies from genre to genre, this correlation gives consistency to the interpretation of the results obtained.

The best results for BC and MZ genres are obtained by the Baseline, applying a single 3-NN classifier to original RelaxCor vectors (F_1-scores: 76.2 and 73.1, respectively). For the rest of the genres, the best results are obtained when the SVD-dimensional representation is used for mention-pairs. An F_1-score of 85 is obtained for the WB genre with a single 3-NN classifier (3-NN + SVD). The proposed approach achieves the best results for two out of the five genres, with an F_1-score of 72.8 for BN and 71.6 for NW, when the SVD-dimensional vectors are classified by the optimized multi-classifier (MultiCl.$_{opt}$ + SVD). Surprisingly, when the optimized dimensions are used, results do not improve (MultiCl.$_{opt}$ + SVD$_{opt}$).

The last column in Table 9 shows the mean accuracy and F_1-scores obtained in each experiment, taking into account the five genres as a whole (the best are shown in bold). The best mean F_1-score is obtained when mention-pairs are classified in the SVD-dimensional vector space by a single 3-NN classifier. In fact, this result is very closely followed by the one obtained in the proposed approach with the multi-classifier, (74.7 and 74.6, respectively). The best mean accuracy is obtained by the proposed approach (72.1). This good results seem to suggest that the dimensions computed by the SVD technique are very appropriate to represent mention-pairs and classify them. Moreover, the use of the multi-classifier system gets to achieve even better results for some of the genres, outperforming the ones obtained by the other classification systems.

7. Conclusions and future work

In this paper a different machine learning approach to deal with the coreference resolution task is presented: a multi-classifier system that classifies mention-pairs in a reduced dimensional vector space created by applying the SVD technique. The approach is tested for OntoNotes, the corpus used in the most recent international challenges such as CONLL-2011 and CONLL-2012, devoted to evaluate coreference resolution systems.

A parameter tuning phase is performed to adjust the dimension of the vector space and the number of classifiers. This optimization process is carried out in an independent way for each genre. Results show different behaviors for the five genres and, therefore, make it difficult to find a general solution and treat the five genres as a unique classification problem.

Three experiments are carried out. In a first experiment, the most similar method to the proposed approach is considered, and results are computed using the original feature vectors and a single 3-NN classifier to set a baseline. A second experiment is performed to measure to what extent working with feature vectors in a reduced dimensional vector space helps to determine coreference resolution of mention pairs. Four single classifiers are applied, being 3-NN the one that obtains the best results. In fact, it outperforms baseline results for three out of the five genres (BN, NW, WB) and is the best for WB genre. In a final experiment, the proposed approach is applied and very promising results are obtained. As a matter of fact, the best results are obtained using it for BN and NW genres.

When mean results per experiment are considered, the SVD-dimensional representation always achieves the best results. This is a very significant fact, because it seems to suggest that the SVD-dimensional representation computed by LSI is a very robust and suitable representation for coreference mention-pairs. The use of such a representation, compared to existing approaches that do not make use of it, may benefit the performance of systems that solve the complete task of mention detection and coreference resolution and, consequently, have an important impact in more general Natural Language Processing tasks that require natural language understanding.

As future work, we plan to experiment with OntoNotes v5.0 Release, the new version available. We also intend to experiment with some other kind of multi-classifer systems. It is important to note that the approach may be applied to corpora in other languages as well.

Acknowledgments

This work was supported by the University of the Basque Country, UPV/EHU, Ikerketaren arloko Errektoreordetza/Vicerrectorado de Investigación.

References

Aha, D.W., Kibler, D., Albert, M.K., 1991. Instance-based learning algorithms. Mach. Learn. 6 (1), 37–66.
Bengtson, E., Roth, D., 2008. Understanding the value of features for coreference resolution. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP '08, Association for Computational Linguistics, pp. 294–303.
Berry, M.W., Browne, M., 2005. Understanding Search Engines: Mathematical Modeling and Text Retrieval. SIAM, Philadelphia, USA.
Berry, M.W., Dumais, S.T., O'Brien, G.W., 1995. Using linear algebra for intelligent information retrieval. SIAM Rev. 37 (4), 573–595.
Björkelund, A., Nugues, P., 2011. Exploring lexicalized features for coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 45–50.
Breiman, L., 1996. Bagging predictors. Mach. Learn. 24 (2), 123–140.
Broscheit, S., Poesio, M., Ponzetto, S.P., Rodriguez, K.J., Romano, L., Uryupina, O., Versley, Y., Zanoli, R., 2010. Bart: a multilingual anaphora resolution system. In: Proceedings of the SemEval-2010, pp. 104–107.
Bryl, V., Giuliano, C., Serafini, L., Tymoshenko, K., 2010. Using background knowledge to support coreference resolution. In: ECAI, Frontiers in Artificial Intelligence and Applications, vol. 215. IOS Press, Amsterdam, The Netherlands, pp. 759–764.
Chang, K.-W., Samdani, R., Rozovskaya, A., Rizzolo, N., Sammons, M., Roth, D., 2011. Inference protocols for coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, pp. 40–44.
Dasarathy, B., 1991. Nearest Neighbor (NN) Norms: NN Pattern Recognition Classification Techniques. IEEE Computer Society Press, Los Alamitos, CA.
Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R., 1990. Indexing by latent semantic analysis. J. Am. Soc. Inf. Sci. 41 (6), 391–407.

Dietterich, T., 1998. Machine learning research: four current directions. AI Mag. 18 (4), 97–136.
Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S., Weischedel, R., 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In: Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC-2004), European Language Resources Association (ELRA), Lisbon, Portugal.
Dumais, S., 2004. Latent semantic analysis. In: ARIST (Annual Review of Information Science Technology), vol. 38. pp. 189–230.
Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H., 2009. The weka data mining software: an update. SIGKDD Explor. Newsl. 11 (1), 10–18.
Hirschman, L., Chinchor, N., 1998. Coreference task definition. In: Proceedings of the Seventh Message Understanding Conference, MUC-7.
Ho, T., Hull, J., Srihari, S., 1994. Decision combination in multiple classifier systems. IEEE Trans. Pattern Anal. Mach. Intell. 16 (1), 66–75.
Kobdani, H., Schütze, H., 2010. Sucre: a modular system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp. 92–95.
Lappin, S., Leass, H.J., 1994. An algorithm for pronominal anaphora resolution. Comput. Linguist. 20 (4), 535–561.
Lee, H., Chang, A., Peirsman, Y., Chambers, N., Surdeanu, M., Jurafsky, D., 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. Comput. Linguist. 39 (4), 885–916.
Manning, C., Bauer, J., Surdeanu, M., Finkel, J., Bethard, S., McClosky, D., 2014. The stanford corenlp natural language processing toolkit. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60.
Mitkov, R., 1998. Robust pronoun resolution with limited knowledge. In: Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, vol. 2, pp. 869–875.
Mitkov, R., 2002. Anaphora Resolution, Studies in Language and Linguistics. Longman, Great Britain.
MUC6, 1995. Coreference task definition. In: Proceedings of the 6th Conference on Message Understanding, MUC-6, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 335–344.
Ng, V., 2010. Supervised noun phrase coreference research: the first fifteen years. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1396–1411.
Nilsson, K., Hjelm, H., 2009. Using semantic features derived from word-space models for swedish coreference resolution. In: Proceedings of the 17th Nordic Conference of Computational Linguistics NODALIDA 2009, vol. 4, Northern European Association for Language Technology (NEALT), Stockholm, Sweden, pp. 134–141.
Nogueira dos Santos, C., Lopes Carvalho, D., 2011. Rule and tree ensembles for unrestricted coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning: Shared Task, Association for Computational Linguistics, Portland, Oregon, USA, pp. 51–55.
Orasan, C., Cristea, D., Mitkov, R., Branco, A., 2008. Anaphora resolution exercise: an overview. In: Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08), European Language Resources Association (ELRA), Marrakech, Morocco.
Pradhan, S.S., Hovy, E.H., Marcus, M.P., Palmer, M., Ramshaw, L.A., Weischedel, R.M., 2007a. Ontonotes: a unified relational semantic representation. Int. J. Semant. Comput. 1 (4), 405–419.
Pradhan, S.S., Ramshaw, L., Weischedel, R.M., MacBride, J., Micciulla, L., 2007b. Unrestricted coreference: identifying entities and events in ontonotes. In: ICSC, IEEE Computer Society, pp. 446–453.
Pradhan, S., Palmer, M., Ramshaw, L., Weischedel, R., Marcus, M., Xue, N., 2011. Conll-2011 shared task: modeling unrestricted coreference in ontonotes. Proceedings of the Fifteenth Conference on Computational Natural Language Learning.
Pradhan, S., Moschitti, A., Xue, N., Uryupina, O., Zhang, Y., 2012. CoNLL-2012 shared task: modeling multilingual unrestricted coreference in OntoNotes. In: Proceedings of the Sixteenth Conference on Computational Natural Language Learning (CoNLL 2012).
Raghunathan, K., Lee, H., Rangarajan, S., Chambers, N., Surdeanu, M., Jurafsky, D., Manning, C., 2010. A multi-pass sieve for coreference resolution. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, EMNLP '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 492–501.
Recasens, M., Màrquez, L., Sapena, E., Martí, M.A., Taulé, M., Hoste, V., Poesio, M., Versley, Y., 2010. Semeval-2010 task 1: coreference resolution in multiple languages. In: Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1–8.
Sapena, E., Padró, L., Turmo, J., 2011. Relaxcor participation in conll shared task on coreference resolution. In: Proceedings of the Fifteenth Conference on Computational Natural Language Learning, CONLL, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 35–39.
Sapena, E., Padró, L., Turmo, J., 2013. A constraint-based hypergraph partitioning approach to coreference resolution. Comput. Linguist. 39 (4), 847–884.
Soon, W.M., Ng, H.T., Lim, D.C.Y., 2001. A machine learning approach to coreference resolution of noun phrases. Comput. Linguist. 27 (4), 521–544.
Uryupina, O., 2010. Corry: a system for coreference resolution. In: Proceedings of the 5th International Workshop on Semantic Evaluation, Association for Computational Linguistics, Uppsala, Sweden, pp. 100–103.
Uryupina, O., Poesio, M., 2012. Domain-specific vs. uniform modeling for coreference resolution. In: Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012), European Language Resources Association (ELRA), Istanbul, Turkey, pp. 187–191.
Versley, Y., Ponzetto, S.P., Poesio, M., Eidelman, V., Jern, A., Smith, J., Yang, X., Moschitti, A., 2008. Bart: a modular toolkit for coreference resolution. In: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies, HLT, Stroudsburg, PA, USA, pp. 9–12.
Zelaia, A., Alegria, I., Arregi, O., Sierra, B., 2005. Analyzing the effect of dimensionality reduction in document categorization for basque. Arch. Control Sci. 15 (4), 703–710.