Coreference Resolution Using Clusterization

Anastasiya Bodrova, Natalia Grafeeva
Saint Petersburg State University
Saint Petersburg, Russia
anastasie.bodrova@gmail.com, n.grafeeva@spbu.ru

***Abstract*—This work describes the experience of creating a coreference resolution system for Russian language. Coreference resolution is a key subtask of Information Extraction, and aims to grouping mentions that refer to the same discourse entity. This work was aimed to applying a clusterization algorithm for Russian-language newswire texts. We narrowed the task to Person proper names clusterization. Our approach model included two steps: mention extraction and clusterization. Mention extraction was proceeded by manually-created grammars for Tomita-parser. For mention grouping, we used agglomerative clusterization on entity level with the help of weighted feature vectors. We run our experiments on newswire texts, annotated for factRuEval-2016 competition, organized by Dialogue Evaluation. We compare our results with competitors. As a baseline, we set built-in Tomita-parser algorithms for name extraction and name clusterization. We got comparable results and outperformed the baseline.**

I. Introduction

Coreference resolution is an important subtask of natural language processing systems [1]. Systems, requiring deep language understanding, such as information extracion or named entity linking, benefit from extracted coreference information.

Coreference resolution is the task of grouping mentions that refer to the same discourse entity. Mentions are text equivalent of Mentions can be named, nominal and pronominal. Table I illustrates an example of mentions of the entity "Joe Smith" [2].

TABLE I. Example of mentions of the entity "Joe Smith"

Named Mention:	Joe Smith, Mr. Smith
Nominal Mention:	the guy wearing a blue shirt
Pronoun Mentions:	he, him

The attention to this task increased, when it had become a separate track in 1995 at MUC-6 (Message Understanding Conference) competition organised by NRAD with the support of DARPA.

Since then, numerous coreference resolution systems have been developed. They commonly used machine learning approaches (both supervised and unsupervised), and improve them with various linguistic and contextual features. The key types of models are mention-pair, mention-ranking and entity-based [3]. The current approaches explore unsupervised methods and entity-based models [4], [5].

In our work, we focus on a problem of applying coreference resolution algorithms for Russian language news. We started from a narrower task of determining clusters of only named mentions, that refer to the same Person-type entity. Actually, it means to group all available name parts (i.e. form of address, personal name, surname, patronymic, nickname) of each person, mentioned in the text. For example, in the novel "War and Peace" by Leo Tolstoy, we will group "prince Andrew", "prince Andrew Nikolaevich", "prince Bolkonski" into one entity "prince Andrew Nikolaevich Bolkonski".

For Russian language the open solution exists for this task, and it is implemented in Tomita-parser (a tool for fact extraction). We took results of its work as a baseline, and tried to build such a coreference resolution system, that can be flexible for changes to specific purposes at each step and can show results not worser than baseline.

We attempted to apply an entity-based approach with agglomerative clustering: each mention starts in its own cluster and then pairs of clusters are merged each step. Merging is proceeded using ranked set of mention pairs and can happen only with absence of contradictions between any two mentions from both clusters.

We run experiments on Dialogue Evaluation factRuEval-2016 datasets of news [6], which were preprocessed for our task, and compared results with the competition participants.

II. Related work

The pre-history of coreference resolution started from a narrower task - anaphora resolution. Anaphora resolution is the task of finding the preceding mentioned name (called antecedent), which some expression (e.g.pronoun or personal name) refers to. For example, in the text "Bob entered the room. He looked confused.", the pronoun "he" refers to the name "Bob". One of the difference between anaphora and coreference, that anaphora resolution resulted in a set of pairs of an expression and its antecedent, and coreference resolution resulted in a clusters of all mentions, that refer to the same discourse entity. Computational theories of discourse, in particular *focusing* and *centering* [7], [8], [9], [10], have heavily influenced coreference research in the 1970s and 1980s, leading to the development of numerous centering algorithms [11].

The work on coreference resolution was initialized at the time of appearence of a separate track in 1995 at MUC-6 (Message Understanding Conference) competition organised by NRAD with the support of DARPA. In 1998, coreference resolution track was also included into MUC-7. These conferences conducted large-scale evaluations and

ISBN 978-952-68397-6-9 (paperback), ISBN 978-952-68397-7-6 (PDF)

developed a considerable amount of annotated data, which stimulate the growth in applying machine-learning methods to the coreference task. Since then three important classes of learning-based coreference models were developed, namely mention-pair model, mention ranking model and entity-based model.

Mention-pair models. The mention-pair model is a classifier that determines, whether a pair of mentions is coreferent. The desicion about coreference on each pair is made independently. The most widely used coreference clustering algorithms are *closest-first clustering* and *best-first clustering* [12], [13], [14], [15], [16].

Entity-based model. Unlike mention-pair models, entity-based models, aims to classify, whether mentions are coreferent, using information from preceding, maybe partially-formed, clusters. It helps to enhance the information for the decision, because it can be not enough information between pair of mentions. Moreover, that model supports transitivity.

The classic example shows the advantage of entity-based model over mention-pair model. For instance, we have a text of three mentions: "Mr.Clinton", "Clinton" and "Hilary Clinton". In mention-pair model, we can independently connect "Mr.Clinton" with "Clinton", and "Hilary Clinton" with "Clinton", and then, due to transitivity, find "Mr.Clinton" and "Hilary Clinton" in one cluster, which is wrong regarding the gender constraints. However, in an entity-based model, if we firslty connect into a cluster "Mr.Clinton" and "Clinton", a link between "Clinton" and "Hilary Clinton" can not appear, because of the gender constraints, which "Clinton" inheritated from a cluster-level features, which, in its turn, come from "Mr.Clinton".

The first attempt of implementing entity-based model was made by Luo et al. [17], who consider all clustering possibilities by searching in a Bell tree representation, and cast the coreference resolution problem as finding the best path from the root node to the leaves.

Other approaches suggested different strategies of optimizing clutering decision: a first-order probabilistic model that allows features based on firs-order logic over a set of mentions [18]; integer linear programming to enforce transitivity [19], [20], [21]; graph partitioning algorithms [22], [23]; using imitation learning and model stacking [5].

Mention ranking models. Mention ranking model is considered as a step between mantion-pair and entity-based models. A ranker allows more than one candidate mention to be examined simultaneously and, by determining which candidate is most probable, it directly captures the competition among them. Ng (2005) made a different use of ranking and recasts the coreference task as ranking candidate partitions generated by different mention-pair systems [24]. Rahman and Ng (2009) proposed a cluster-ranking approach that combines the strengths of mention rankers and entity-mention models [25].

Unsupervised models. The applying of unsupervised models is motivated by the absence or costliness of annotated data.

The first attempt was made by Cardie and Wagstaff in 1999, their approach applies clustering on feature vectors, that represent mentions.

Researchers are still investigating the abilities of that type of model. [26] presented a mention-pair nonparametric fully-generative Bayesian model for unsupervised coreference resolution. Based on this model, [27] probabilistically induced coreference partitions via EM clustering. [28] proposed an entity-mention model that is able to perform joint inference across mentions by using Markov Logic. [29] proposed a generative, unsupervised ranking model for entity coreference resolution.

III. Mention extraction

In this paper we consider mentions as person proper names [30]. Personal proper names can consist of personal name, family name, patronymic name or nickname. We used this classification due to standard russian structure of the full name [31]. It fits most slavonic languages, however other languages can contain extra parts such as title, middle name, matronymic name. So in our work we do not regard cultural properties while parsing these names, but consider them as a personal name and a family name. For example, arabic name *Muhammad ibn Salman ibn Ameen Ahl-Farsi* actually means *Muhammad, son of Salman, son of Ameen, the Persian* will be interpreted as personal name *Muhammad* and family name *ibn Salman ibn Ameen Ahl-Farsi*.

A. Tool

For mention detection, we used a rule-based tool - Tomita-parser [32], an instrument for extracting structured data from the natural language texts. It is based on the GLR-parser algorithm [33] and uses the formalism of context-free grammars. Tomita-parser analyzes the text, using linguistic-based grammars, which consist of a set of rules and linked-in gazeeteers (kind of inner key-word vocabularies). Table II illustrates an example of a rule and a substring, which fits it.

TABLE II. Example of a Tomita-parser rule. *Person* - nonterminal for full person names. *kwtype<address>* - any word from key-word vocabulary. *nob_part* - nonterminal for nobiliary particles. *Word<surmame>* - terminal with the grammar feature for surnames

Person ->	kwtype<address>	nob_part	Word<surname>
	Dame	de	Monsoreau
	Графиня	де	Монсеро

B. Text processing

Mention extraction step is aimed to getting a set of mentions with features from an input plain text. To achieve this, we use two grammars, written for Tomita-parser. The first one is responsible for preprocessing, with its help we extract all non-dictionary words and create temporary vocabularies of names. The second one, using created vocabularies, gathers parts of the name together and extract corresponding features. All grammars use manually created key words vocabularies, including stopwords, some

geographical entites, addresses and other additional information.

This model is shown at Fig.1

1) Non-dictionary words extraction grammar: Tomita-parser uses inner dicionary, which includes some personal names. However, it is impossible to cover all existing names, that can appear in the text, but non-dictionary names can be extracted by the grammar rules.

The main issue of this step lies in separation names, that refer to persons, from names, that refer to other types of entities (e.g. organization, location). To achieve this challenge, we write the rules, that describe the context with a high probability to refer a specific entity-type. For example, we use keywords vocabularies with location and organization descriptors.

If any non-dictionary name appears in appropriate context, it gets label "pers_check", for person context, or "misc_check", for other contexts. Those names, that are labeled as person names, are added to a created temporary vocabulary, others are added to a stop-list vocabulary. As a result we received a vocabulary, that will be plugged in into the next grammar.

2) Full name extraction grammar: After getting non-dictionary words vocabulary, we can form full name from its smaller parts: Firstname, Lastname, Patronymic name, and, additionally, non-dictionary name, that can be any of aforementioned parts. Moreover, we extract some extra attributes, such as Gender, Address and Descriptor, which are described below.

While extracting, all names are normalized using built-in Tomita-parser algorithm. Normalization is the process of transforming the word into canonical form (to the nominative case), e.g. *"Борису"* -> *"Борис"* (*"Borisu"* -> *"Boris"*) .

As a result we get a set of mentions with attributes.

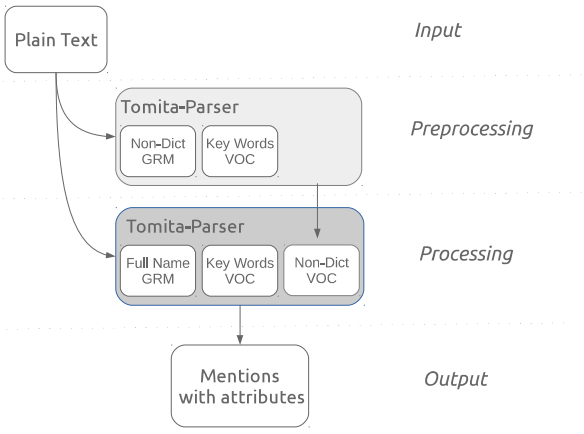

Fig. 1. Mention Extraction Model

3) Mention attributes: Each mention is represented by a set of features, described below. The distribution of features is presented at Fig.2.

- **NameNotNorm**
  This attribute just keep the string as it is appeared in the text.
- **Firstname**
- **Lastname**
- **Patronymic**

  Firstname, Lastname and Patronymic attributes are filled, using the inner Tomita-parser dictionary. For all non-dictionary words we tagged attributed, based on their lexical or syntactic features. For example, if an extracted word ends with *"швили"* (*"shvili"*) - it is a high probability for a word to be a Georgian surname, or if word goes before the patronymic name, it will be a firstname.
- **Gender**
  The Gender attribute can equal *"male"*, *"female"* or *"dual"*. The gender is detected in several ways. Firstly, we relay on grammatical gender (if available) from the Tomita-parser inner dictionary. Secondly, we use the vocabulary of addresses, which is manually sorted by gender. And then, we look at grammatical gender in syntactically dependent verbs, adjectives and descriptors. If there os not enough information for detecting gender, than attribute is *"dual"*.
- **Address**
  The address attribute is can be filled with any word from address vocabulary. Address is a noun phrase, that is used as an additional part to the name, while appealing to person (e.g. *"mister"*, *"duke"*).
- **Descriptor**
  Desciptor is a noun phrase, that descibes some class, a person belongs (e.g. job, kinship role) About 80% of descriptors in newswire texts appear with the first mention of a person. It relates with an attempt to explain, whom the text is about. Hence, there is a higher probability to meet a name with a descriptor before the same person's name without it.

The illustation example of mention extraction process is shown below.

Input text:
В США содержатся в заключении без обвинений два журналиста: фотограф Ассошиэйтед Пресс Билал Хассеин (Bilal Hussein), и оператор Аль-Джазиры Сами аль-Хаджж (Sami al-Haj), которые находится в тюрьме уже пять лет и в настоящий момент содержится в Гуантанамо (Куба).

Non-Dictionary Word Extraction Grammar Output:
В США содержатся в заключении без обвинений два журналиста: фотограф ***[Ассошиэйтед]*** *Пресс Билал* ***[Хассеин]*** *(Bilal Hussein), и оператор Аль-Джазиры* ***[Сами]*** ***[аль-Хадж]*** *(Sami al-Haj), которые находится в тюрьме уже пять лет и в настоящий момент содержится в Гуантанамо (Куба).*

RawName
{
Name = Ассошиэйтед
NameNotNorm = Ассошиэйтед
Misc_check = true
}
RawName
{
Name = Хассеин
NameNotNorm = Хассеин
Pers_check = true
}
RawName
{
Name = Сами
NameNotNorm = Сами
Pers_check = true
}
RawName
{
Name = аль-Хадж
NameNotNorm = аль-Хадж
Pers_check = true
}

Generated Vocabularies:

Miscellaneous: ассошиэйтид

Names: хассеин, сами, аль-хадж

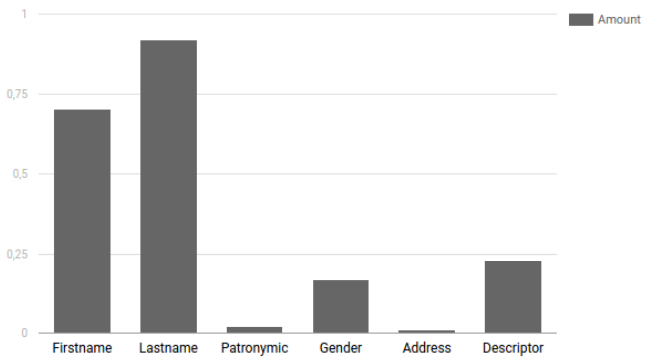

Fig. 2. Distribution of Mention Attributes in Development Set

Full Name Extraction Grammar Output:
В США содержатся в заключении без обвинений два журналиста: фотограф Ассошиэйтед Пресс ***[Билал Хассеин]*** *(Bilal Hussein), и оператор Аль-Джазиры* ***[Сами аль-Хадж]*** *(Sami al-Haj), которые находится в тюрьме уже пять лет и в настоящий момент содержится в Гуантанамо (Куба).*

StrictName
{
NameNotNorm = Билал Хассеин
Firstname = Билал
Lastname = Хассеин
Patronymic = None
Gender = male
Address = None
Descriptor = фотограф Ассошиэйтед Пресс
}
StrictName
{
NameNotNorm = Сами аль-Хадж
Firstname = Сами
Lastname = аль-Хадж
Patronymic = None
Gender = male
Address = None
Descriptor = оператор Аль-Джазиры
}

Mentions with Attributes:
1: (’Билал’, ’Хассеин’, ’None’, ’male’, ’None’, ’фотограф Ассошиэйтед Пресс’)
2: (’Сами’, ’аль-Хадж’, ’None’, ’male’, ’None’, ’оператор Аль-Джазиры’)

IV. Clusterization

A. Denotations and definitions

Let us consider each document $\mathcal{D}$ as a set of n mentions $\mathcal{M} = \{m_1, ...m_n\}$. All mentions form a set of unique mention pairs $\mathcal{MP} = \{(i,j) \mid 1 \leq i < j \leq n\}$, where each mention is represented with its index.

Each mention pair from $\mathcal{MP}$ has its pairwise score $\mathcal{P}_{(i,j)\in\mathcal{MP}}$, counted by function $score(i,j)$, that shows how close the mentions are to be coreferent:

$$score(i,j) = \begin{cases} None, & if Disjoint(i,j) \\ \theta^T f(i,j), & otherwise \end{cases}$$

where $f(i,j)$ is a feature vector of two mentions, and θ is a feature weight vector, which is got by training a mention pair model on development set (IV-B). Disjoint function tells, if two mentions have a contradiction, which prevent their presence in one cluster. It takes takes the following form:

$$Disjoint(i,j) = \begin{cases} true, & if\ Firstname\_match[Dismatch] \\ & or\ Lastname\_match[Dismatch] \\ & or\ Gender\_match[Dismatch] \\ & or\ \theta^T f(i,j) < 0 \\ false, & otherwise \end{cases}$$

To denote the goal of coreference resolution, let us consider $n \times n$ Boolean triangle matrix $\mathcal{C}$ as a result of *clustering*, where $\mathcal{C}_{\rangle|} = 1$ if m_i and m_j are coreferent, and 0 otherwise. Clustering in matrix $\mathcal{C}$ is *valid* if and only if the relevant entries satisfy the transitivity constraint: $(\mathcal{C}_{ij} = 1 \wedge \mathcal{C}_{ij} = 1) \Rightarrow \mathcal{C}_{ik} = 1\ \forall\ 1 \leq i < j < k \leq n$. Let denote $C[i]$ as a cluster, that i-th mention belongs to.

The goal of coreference resolution is to provide a *valid clustering* $\mathcal{C}$ on document $\mathcal{D}$. The clusterization step is described at IV-D.

B. Mention pair model

We train our mention pair model using logistic classifier to learn feature weights. Mention pair model predicts, whether two mentions belong to the same cluster. The probability of coreference takes the standard logistic form:

$$p_\theta(i,j) = (1 + e^{-\theta^T f(i,j)})^{-1}$$

where $f(i,j)$ is a feature vector on m_i and m_j and θ is a feature weight vector, we wish to learn.

We consider $\mathcal{M}$ as the set of all mentions in the development set, let $\mathcal{T}(j)$ denote the set of mentions indexes, preceeding j, such that m_i and m_j are coreferent, and $\mathcal{F}(j)$ - such that m_i and m_j are not coreferent. The sets $\mathcal{T}(j)$ and $\mathcal{F}(j)$ formed within document $\mathcal{D}$, the mention m_j belongs to. We want to find a parameter vector θ that assigns high probabilities to the candidate mentions in $\mathcal{T}(j)$ and low probabilities to the ones in $\mathcal{F}(j)$.

The model is trained on our development set by solving the following optimization problem using L1-regularization:

$$\mathrm{L}(\theta) = -\sum_{m \in \mathcal{M}} \Big(\sum_{t \in \mathcal{T}(m)} log\left(p_\theta\left(t,m\right)\right) + \\ + \sum_{f \in \mathcal{F}(m)} \left(log\left(1 - p_\theta\left(f,m\right)\right)\right) + min\left\|\theta\right\|_1 \Big)$$

C. Feature vector

Feature vector represents matches between different attributes of two mentions. Attributes **Firstname**, **Lastname** and **Patronymic** have five variants of match:

- *Strict Match*
  Attributes' values are identical.
- *Relaxed Match*
  Attributes' values are close to be identical. For relaxed match check we count the Levenstein Distance between the mentions' strings. Levenstein Distance is a string metric for measuring the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.
  Relaxed match check is aimed to find similar names despite the errors of normalization and text misprints. We allow Levenstein Distance to be less than 3, because it is the maximum length of noun suffix in Russian language.
  Relaxed match is also equals 'true', if one string is a part of another, e.g. *'Anna Maria'* and *'Anna'*.
- *First Letter Match*
  If at least one of the attributes' values is a shortened to one letter and this letter is equal to the first letter of another value.
- *None Field*
  At least one of attributes' values is *'None'*.
- *Dismatch*
  Attributes' are different at least in one word.

Attributes **Gender** and **Address** has only match variants: Match, Dismatch and None Field. Gender and Address attributes can be equal to a limited list of dictionary words, that is why we do not deal with normalization errors and misprints.

Attribute **Descriptor** reflects only the presence the descriptor in mentions.

D. Clusterization algorithm

Clusterization is proceeded with the help of mention pair scores and transitivity restriction.

A widely used clusterization approach for coreference resolution is *best-first clustering* [14]. For each mention, the best-first algorithm assigns the most probable preceding mention, that can be coreferent to it.

The weakness of this approach appeares in making decisions on local, pairs level, not entity one. That is why, we can get a cluster with conflict elements, e.g.[*Hilary*

TABLE III. FEATURES ON MENTION PAIRS

ID	Feature	Values	Example
1	Firstname_match	1. Strict Match 2. Relaxed Match 3. First Letter Match 4. None Field 5. Dismatch	"Андрей" "Андрей" "Андрею" "Андрей" "Андрей" "А." "Андрей" "None" "Андрей" "Мария"
2	Lastname_match	1. Strict Match 2. Relaxed Match 3. First Letter Match 4. None Field 5. Dismatch	"Иванов" "Иванов" "Иванову" "Иванов" "Иванов" "И." "Иванов" "None" "Иванов" "Петров"
3	Patr_match	1. Strict Match 2. Relaxed Match 3. First Letter Match 4. None Field 5. Dismatch	"Иванович" "Иванович" "Иванович" "Ивановичу" "Иванович" "И." "Иванович" "None" "Иванович" "Петрович"
4	Gender_match	1. Match 2. None Field 3. Dismatch	"female" "female" "female" "None" "female" "male"
5	Addr_match	1. Match 2. None Field 3. Dismatch	"мистер" "мистер" "мистер" "None" "князь" "король"
5	Descr_presence	1. Both true 2. First is true 3. Second is true 4. Both false	"true" "true" "true" "false" "true" "true" "false" "false"

Clinton, Clinton, Bill Clinton]. That happens because the coreference decision between *Hilary Clinton* and *Clinton* makes independently of the one between *Bill Clinton* and *Clinton*.

To avoid this problem, we support an entity level at merging step by checking, if score function of any possible pair from two clusters, intended to merge, do not return *'None'*, i.e. they are strictly not coreferent.

We use an agglomerative clustering for our approach: each mention starts in its own single-element cluster, and then, at each step two clusters are merged.

Firstly, we sort all mention pairs in descending order according to their pairwise scores. This causes clustering to occur in an easy-first fashion, where harder decisions are delayed until more information is available.

Then we iterate through the sorted list of pairs in order. For each pair, we make a binary desicion on whether or not the clusters containing these pairs should be merged. The function $CanMerge$ returns *true* if there are no contradictions between any two mentions from both clusters, and $false$ otherwise. Algorithm 1 shows the procedure.

Algorithm 1 Agglomerative Clustering

Input: Pairwise score set $\mathcal{P}$
Output: Clustering $\mathcal{C}$

$\mathcal{S} \leftarrow Sort_{desc}(\mathcal{P}_{(i,j)} \mid score(i,j) \neq None)$;
$for\ (i,j) \in \mathcal{S}$:
  $if\ \mathcal{C}[i] \neq \mathcal{C}[j]$:
    $if\ CanMerge(\mathcal{C}[i], \mathcal{C}[j])$:
      $Merge(\mathcal{C}[i], \mathcal{C}[j])$

V. Experiments

We run our experiments on the materials of factRuEval-2016 competition, organized by Dialogue Evaluation command [6]. The materials consist of 254 annotated newswite texts, 122 of them were given as a development set, and others, 132, were used for evaluation (Table IV). In order to compare our results with competitors, we improve our model only on the development set.

The competition had three tracks (Named Entity Extraction, Coreference Resolution, Fact Extraction). So, we can compare separately our results on mention detection and coreference resolution.

Also we can compare our results with built-in Tomita-parser algorithm for names extraction and names clusterization.

TABLE IV. Statistics on datasets

	Development Set	Test Set
#Doc	122	132
#Mentions	741	1388
#Entities	391	643
#Avg.Mentions	6.07	10.5
#Avg.Entities	3.22	4.87

Algorithms performance has been measured using traditional evaluation metrics Precision, Recall and F_1-measure (or just F-measure). Precision is the fraction of retrieved instances that are relevant, while Recall is the fraction of relevant instances that are retrieved. F-measure is the harmonic mean of Precision(P) and Recall(R): $2 \cdot \frac{P \cdot R}{P+R}$.

All results are sorted by F-measure.

A. Results on mention extraction

Mention Extraction was the first task of factRuEval competition, in which 13 teams participated in (their names are encrypted using the color names in the table). We can compare our results with participants only on test set.

Tomita-parser provide a built-in algorithm for extraction named mentions. It finds names from inner dictionary in the text and combine those, that follow each other and has gender and number agreement. We use results of Tomita-parser as a baseline.

The results of comparison on test set are shown at Table V and Fig.3.

Comparing our approach results to Tomita-parser ones, we can say, that Tomita has a higher precision, but lower recall. That happens because Tomita uses only inner dictionaries and do not predicts, whether unknown words are names or not, unlike in our approach. So, Tomita loses recall becase of absence of predictions, and our approach loses precision because of wrong predictions.

B. Results on clusterization

The second task of factRuEval competition was dedicated to uniting mentions into the entities, they refer to, and only 5 teams decided to take part in this task. The results of comparison on test set are shown at Table VI and Fig.4.

Our approach is not close to the top results, that are taken mostly by commercial companies in natural language processing sphere. Firstly, the reason of it in dependence of coreference resolution results on mention extraction ones, and errors at the first step continue at the second one. Secondly, commercial companies have more complicated

TABLE V. Mention Extraction Results on Test Set

	Precision	Recall	F-measure
violet	0.9450	0.9155	0.9300
crimson	0.9620	0.8829	0.9208
black	0.9114	0.9236	0.9175
pink	0.9636	0.8677	0.9132
aquamarine	0.9080	0.9174	0.9127
grey	0.9556	0.8726	0.9122
our approach	0.8964	0.9131	0.9047
beige	0.9274	0.8785	0.9023
brown	0.9608	0.8395	0.8961
purple	0.8972	0.8805	0.8888
green	0.9300	0.8403	0.8829
orange	0.9486	0.7861	0.8597
tomita-parser name-extraction algorithm	0.9428	0.7862	0.8574
white	0.9537	0.7399	0.8333
ruby	0.9169	0.7270	0.8110

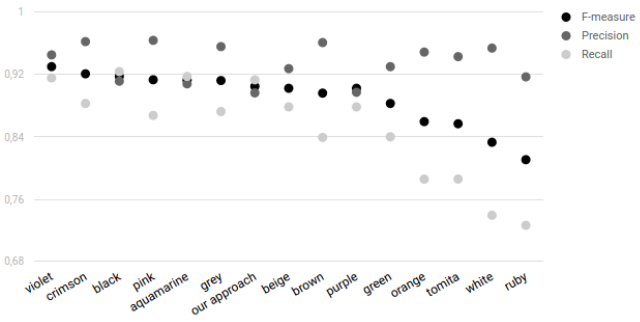

Fig. 3. Mention Extraction Results

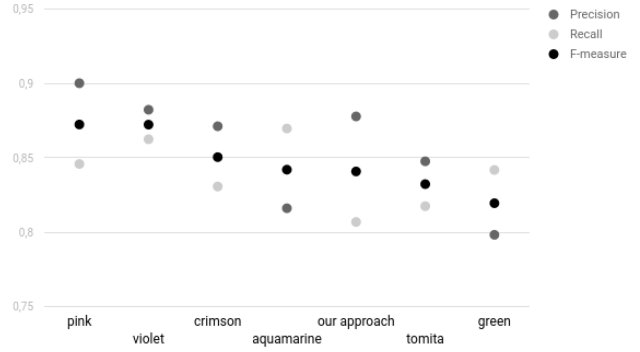

Fig. 4. Clusterization Results

engines, e.g. they are able to analyze text on syntax level or have bigger dictionaries.

VI. Conclusion

In this article we described our experience in bulding a coreference resolution system for Russian language.

Coreference resolution consists of two steps: mention extraction and clusterization. Mention extraction was proceeded using grammars, written for Tomita-parser. For clusterization algorithm we chose agglomerative clustering on entity-level and weighted pairwise features.

Experiments shows, that we have got comparable results, which outperfotm the baseline, the built-in Tomita-parser's algorithms. We plan to improve our work with other entity-types (location, organization), other mention types (nominal and pronominal) and other text types (e.g. fiction).

TABLE VI. Clusterization Results on Test Set

	Precision	Recall	F-measure
pink	0.9006	0.8459	0.8724
violet	0.8823	0.8625	0.8723
crimson	0.8712	0.8308	0.8505
aquamarine	0.8162	0.8697	0.8421
our approach	0.8778	0.8070	0.8409
tomita-parser name-clusterization algorithm	0.8477	0.8176	0.8324
green	0.7984	0.8419	0.8196

The project code is open and available at: https://github.com/lasveritas/coreference-resolution.

References

[1] Ruslan Mitkov, Richard Evans, Constantin Orasan, Iustin Dornescu, and Miguel Rios. Coreference resolution: To what extent does it help nlp applications? In *TSD*, Lecture Notes in Computer Science, pages 16–27. Springer, 2012.

[2] Linguistic Data Consortium. Ace (automatic content extraction) english annotation guidelines for entities. 2008.

[3] Vincent Ng. Supervised noun phrase coreference research: The first fifteen years. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1396–1411, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[4] Xuezhe Ma, Zhengzhong Liu, and Eduard Hovy. Unsupervised ranking model for entity coreference resolution. In *Proceedings of NAACL-2016*, San Diego, California, USA, June 2016.

[5] Kevin Clark and Christopher D. Manning. Entity-centric coreference resolution with model stacking. In *Association for Computational Linguistics (ACL)*, 2015.

[6] "factrueval2016", https://github.com/dialogue-evaluation/factrueval-2016/tree/master/devset, 2016.

[7] Barbara J. Grosz. The representation and use of focus in a system for understanding dialogs. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, pages 67–76, 1977.

[8] Candace L Sidner. Towards a computational theory of definite anaphora comprehension in english discourse. Technical report, Cambridge, MA, USA, 1979.

[9] Barbara J. Grosz, Scott Weinstein, and Aravind K. Joshi. Centering: A framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225, June 1995.

[10] Barbara J. Grosz, Aravind K. Joshi, and Scott Weinstein. Providing a unified account of definite noun phrases in discourse. In *Proceedings of the 21st Annual Meeting on Association for Computational Linguistics*, ACL '83, pages 44–50, Stroudsburg, PA, USA, 1983. Association for Computational Linguistics.

[11] Marilyn Walker and Ellen Joshi, Aravind ans Prince. *Centering Theory in Discourse*. Oxford University Press, 1998.

[12] Wee Meng Soon, Daniel Chung Yong Lim, and Hwee Tou Ng. A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics*, 27(4):521–544, 2001.

[13] Michael Strube, Stefan Rapp, and Christoph Müller. The influence of minimum edit distance on reference resolution. In *Proceedings of Empirical Methods in Natural Language Processing Conference*, pages 312–319, 2002.

[14] Vincent Ng and Claire Cardie. Improving machine learning approaches to coreference resolution. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 104–111, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[15] Xiaofeng Yang and Jian Su. Coreference resolution using semantic relatedness information from automatically discovered patterns. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 528–535, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[16] Eric Bengtson and Dan Roth. Understanding the value of features for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 294–303, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[17] Xiaoqiang Luo, Abe Ittycheriah, Hongyan Jing, A Kambhatla, and Salim Roukos. A mention-synchronous coreference resolution algorithm based on the bell tree. In *Proceedings of the ACL*, pages 135–142, 2004.

[18] Aron Culotta, Michael Wick, Robert Hall, and Andrew McCallum. First-order probabilistic models for coreference resolution. In *Proceedings of HLT-NAACL 2007*, 2007.

[19] Pascal Denis and Jason Baldridge. Joint determination of anaphoricity and coreference resolution using integer programming. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 236–243, Rochester, New York, April 2007. Association for Computational Linguistics.

[20] Jenny Rose Finkel and Christopher D. Manning. Enforcing transitivity in coreference resolution. In *Proceedings of ACL-08: HLT, Short Papers*, pages 45–48, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[21] Xiaofeng Yang, Jian Su, Jun Lang, Chew Lim Tan, Ting Liu, and Sheng Li. An entity-mention model for coreference resolution with inductive logic programming. In *Proceedings of ACL-08: HLT*, pages 843–851, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[22] Andrew Mccallum and Ben Wellner. Toward conditional models of identity uncertainty with application to proper noun coreference. In *NIPS*, pages 905–912. MIT Press, 2003.

[23] Cristina Nicolae and Gabriel Nicolae. Bestcut: A graph algorithm for coreference resolution. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 275–283, Sydney, Australia, July 2006. Association for Computational Linguistics.

[24] Vincent Ng. Machine learning for coreference resolution: From local classification to global ranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 157–164, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[25] Altaf Rahman and Vincent Ng. Narrowing the modeling gap: A cluster-ranking approach to coreference resolution. *J. Artif. Int. Res.*, 40(1):469–521, January 2011.

[26] Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[27] Vincent Ng. Unsupervised models for coreference resolution. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 640–649, Honolulu, Hawaii, October 2008. Association for Computational Linguistics.

[28] Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with markov logic. 2008.

[29] Xuezhe Ma, Zhengzhong Liu, and Eduard Hovy. Unsupervised ranking model for entity coreference resolution. In *Proceedings of NAACL-2016*, San Diego, California, USA, June 2016.

[30] Wikipedia. Proper noun — wikipedia, the free encyclopedia, 2016. [Online; accessed 24-April-2016].

[31] Wikipedia. Eastern slavic naming customs — wikipedia, the free encyclopedia, 2016. [Online; accessed 24-April-2016].

[32] Tomita-parser, yandex technologies, https://tech.yandex.ru/tomita/?ncrnd=2315.

[33] Alon Lavie and Masaru Tomita. Glr* - an efficient noise-skipping parsing algorithm for context free grammars. In *In Proceedings of the Third International Workshop on Parsing Technologies*, pages 123–134, 1993.