ANAPHORA ANALYSIS BASED ON ABBYY COMPRENO LINGUISTIC TECHNOLOGIES

Bogdanov A. V. (abogdanov@abbyy.com),
Dzhumaev S. S. (sdzhumaev@abbyy.com),
Skorinkin D. A. (dskorinkin@abbyy.com),
Starostin A. S. (astarostin@abbyy.com)

ABBYY, Moscow, Russia

This paper presents an anaphora analysis system that was an entry for the Dialog 2014 anaphora analysis competition. The system is based on ABBYY Compreno linguistic technologies. For some of the tasks of this competition we used basic features of the Compreno technology, while others required building new rules and mechanisms or making adjustments to the existing ones. Below we briefly describe the mechanisms (both basic and new) that were used in our system for this competition.

Key words: anaphora resolution, coreference resolution, syntactic analysis, syntactic-semantic analysis

Introduction

The main task of the ABBYY Compreno system is to convert the input text into a semantic structure that is a tree where nodes are concepts and arcs are relations between these concepts. For details see [1] and [4].

At the early stage of the analysis process the structure of a sentence is represented as a syntactic tree. The syntactic analysis of the input text is complete, i.e. every item of the input text takes some syntactic slot of some parent.

Then the syntactic tree is augmented with non-tree links. While tree links encode syntactic dominance, non-tree links capture conjunction, pronominal anaphora, PRO control, and other non-local dependencies between nodes.

Further follows the transition from syntactic to semantic structure. During this process every parent-child arc in the tree is interpreted, and each node gets a semantic role related to its parent. The switch from syntactic slots to semantic roles is possible because each lexeme has a diathesis description—a list of correspondences between the syntactic slots that can connect to it and their semantic roles. During this transition the nodes that were bound with a non-tree link are replaced with their controllers. Let us consider an example:

(1a) Input text
Мальчик дал девочке свое яблоко.

(1b) Syntactic tree without non-tree links

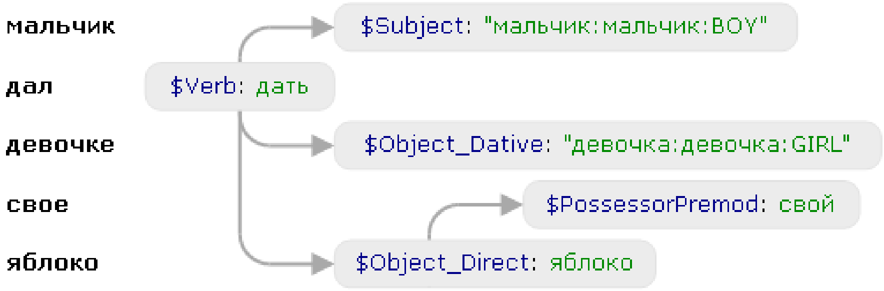


Fig. 1. Syntactic tree without non-tree links

(1c) Semantic tree with non-tree links

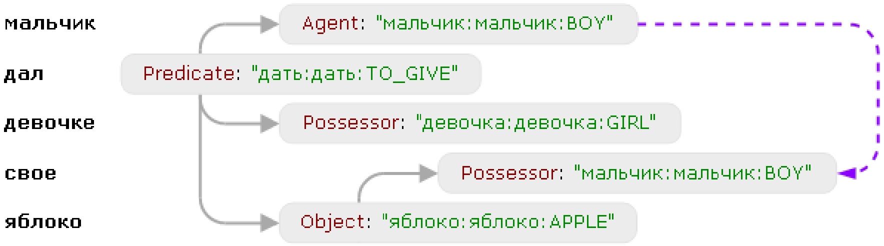

мальчик
дал
девочке
свое
яблоко
Agent: "мальчик:мальчик:BOY"
Predicate: "дать:дать:TO_GIVE"
Possessor: "девочка:девочка:GIRL"
Possessor: "мальчик:мальчик:BOY"
Object: "яблоко:яблоко:APPLE"

Fig. 2. Semantic tree with non-tree links

In (1c, fig. 2) the node *свое* is replaced with its non-tree controller *мальчик* which takes a semantic role of Possessor. If a controller or pronoun parent belonged to some other lexical class, its semantic role could be different. For example:

(2a) Input text
Мальчик знает своего врага.

(2b) Syntactic tree without non-tree links

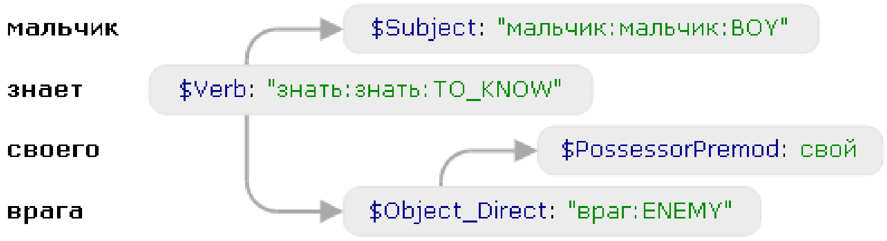

мальчик
знает
своего
врага
$Subject: "мальчик:мальчик:BOY"
$Verb: "знать:знать:TO_KNOW"
$PossessorPremod: свой
$Object_Direct: "враг:ENEMY"

Fig. 3. Syntactic tree without non-tree links

(2c) Semantic tree with non-tree links

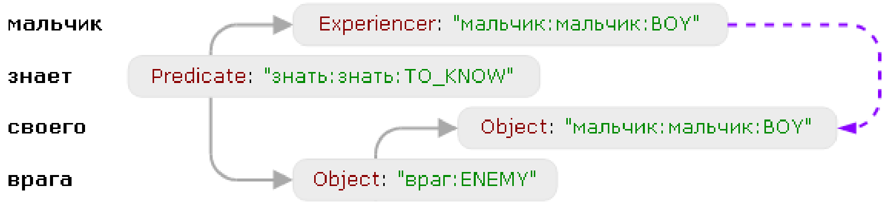

мальчик
знает
своего
врага
Experiencer: "мальчик:мальчик:BOY"
Predicate: "знать:знать:TO_KNOW"
Object: "мальчик:мальчик:BOY"
Object: "враг:ENEMY"

Fig. 4. Semantic tree with non-tree links

In (2) one can see that the same non-tree link as in (1) (between a Subject and a reflexive pronoun) results in a different semantic relation between the controlled node and its parent (semantic role of Object). This happens because when the controlled node is replaced with its controller, the semantic role is chosen depending on the lexical classes of both the controller and the node's parent. If more than one semantic role is possible for a given pair of items all of the possibilities are estimated and the best of them is chosen.

This mechanism of choosing semantic roles for the controlled node also helps us choose the most convenient controller for a given node, as demonstrated below.

1. Anaphora

1.1. Pronominal anaphora

One of the types of non-tree links in the Compreno system is pronominal anaphora. Pronominal anaphora resolution is an existing feature of the system, and therefore we did not have to build any special mechanisms for the purposes of the competition.

The pronominal anaphora rules are triggered if the system finds certain pronouns in the input text. Among such pronouns are: *он, она, оно, они, я, мы, ты, вы, себя, свой, друг друга, таковой* and some others. Each pronominal anaphora rule consists of the following components:

(3)

- list of pronouns that trigger the rule
- description of possible paths (via syntactic slots) from a possible controller to a pronoun
- description of possible properties of a controller
- a rule of agreement between a controller and a pronoun
- linear direction of the link (whether controller is to the left of the pronoun or to the right)
- value of the link

For example in (1) the appropriate rule chooses the Subject node as a controller because there is no path from the Dative object in this rule and there is a path from the Subject.

A description of possible properties of a controller is used to exclude controllers that are obviously impossible, for example such non-referential noun phrases as *в 2014 году, в трактористы, с моей точки зрения, в одностороннем порядке, по его требованию* etc.

In unambiguous examples like *Мальчик любит девочку. Она красивая*. the appropriate rule will choose *девочку* as a controller due to the agreement rule which says that in this anaphora rule a controller must have the same gender and number as a pronoun.

Now let us take an ambiguous example:

(4a) Input text
Мальчик любит этот дом—он его строил.

At the early stage of the analysis process we have a syntactic tree as follows:

(4b) Syntactic tree without non-tree links

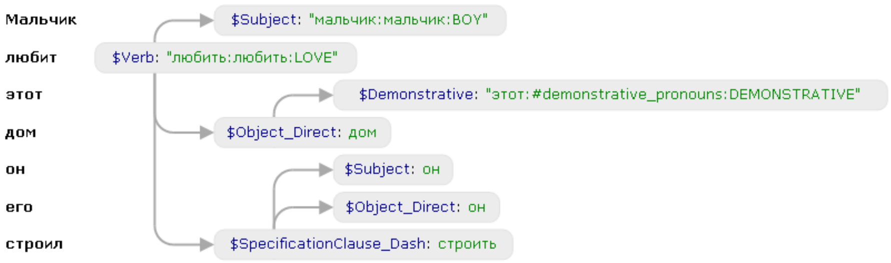

Мальчик
любит
этот
дом
он
его
строил
$Subject: "мальчик:мальчик:BOY"
$Verb: "любить:любить:LOVE"
$Demonstrative: "этот:#demonstrative_pronouns:DEMONSTRATIVE"
$Object_Direct: дом
$Subject: он
$Object_Direct: он
$SpecificationClause_Dash: строить

Fig. 5. Syntactic tree without non-tree links

Then, as the pronouns (*он, его*) are found in the text, the anaphora rules are triggered and produce following links:

(4c) link 1:
Proform "он"; ProformParent "строить"; ProformSlot Object_Direct; Controller "дом"
link 2:
Proform "он"; ProformParent "строить"; ProformSlot Object_Direct; Controller "мальчик"
link 3:
Proform "он"; ProformParent "строить"; ProformSlot Subject; Controller "мальчик"
link 4:
Proform "он"; ProformParent "строить"; ProformSlot Subject; Controller "дом"

Then all possible sets of the non-tree links are formed (in every set, for one pronoun there is no more than one controller, which means that a pronoun may not have a controller) and for each set the system seeks to replace a pronoun with its controller and choose a semantic role for it. It gives us a set of possible syntactic structures with replaced pronouns. These structures are ranked depending on the semantic compatibilities of all the items in given semantic roles (for details on the semantic compatibility and its evaluation see [4]). The best structure is chosen as a result of the analysis.

(4d) Semantic tree with non-tree links

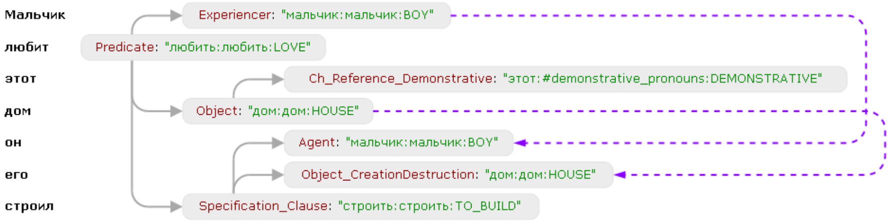

Мальчик
любит
этот
дом
он
его
строил
Experiencer: "мальчик:мальчик:BOY"
Predicate: "любить:любить:LOVE"
Ch_Reference_Demonstrative: "этот:#demonstrative_pronouns:DEMONSTRATIVE"
Object: "дом:дом:HOUSE"
Agent: "мальчик:мальчик:BOY"
Object_CreationDestruction: "дом:дом:HOUSE"
Specification_Clause: "строить:строить:TO_BUILD"

Fig. 6. Semantic tree with non-tree links

In (4d, fig.6) one can see that the pronoun *он* is replaced with its controller *мальчик,* which takes the semantic role of Agent. In its turn, the pronoun *его* is replaced with its controller *дом*, which takes the semantic role of Object_CreationDestruction.

That is how semantic compatibility between possible controllers and pronoun parents helps us in anaphora resolution.

1.2. Relative anaphora

Another type of non-tree links that is used in the Compreno system and was included in our competition links set is relative anaphora. By this term we mean a link between a noun phrase and a relative pronoun of a relative clause governed by this noun phrase like in example *Мальчик, который пришел*.

Links of this type are also drawn by special rules which have almost the same components as in (3) except that relative pronouns, unlike personal pronouns, must always be controlled, i.e. if for a given relative pronoun a controller is not found, then the whole structure is considered invalid. In semantic structure relative pronouns are also replaced with their controllers and choose appropriate semantic roles, which also helps choose the best controller among possible candidates relying on semantic compatibility.

Of course, a range of possible controllers in this case is much narrower than in the previous one, because a controller of a relative pronoun must govern its relative clause, and this information is stored in a corresponding rule as a description of possible paths between a controller and a pronoun. But even relative anaphora may have ambiguous cases, such as:

(5a) Input text
Мальчик видит игрушку девочки, которая пришла.

In (5a) for disambiguation the system should recognize that a girl is more likely to be able to walk than a toy. And this information can be obtained only from the semantic compatibility between a controller and pronoun parent. So for this sentence the semantic tree looks as follows:

(5b) Semantic tree with non-tree links

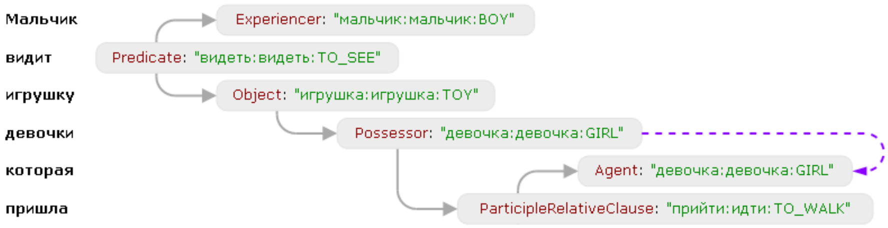

Мальчик
видит
игрушку
девочки
которая
пришла
Experiencer: "мальчик:мальчик:BOY"
Predicate: "видеть:видеть:TO_SEE"
Object: "игрушка:игрушка:TOY"
Possessor: "девочка:девочка:GIRL"
Agent: "девочка:девочка:GIRL"
ParticipleRelativeClause: "прийти:идти:TO_WALK"

Fig. 7. Semantic tree with non-tree links

In (5b, fig. 7) relative pronoun *которая* is replaced with its controller *девочка*, which takes a semantic role of Agent. The structure where *игрушка* takes some semantic role of *прийти* was also considered, but was dismissed as having lower value.

2. Coreference

As challenging as it is, pronominal anaphora nevertheless represents only a limited subclass of the reference phenomena. Full-scale coreference resolution requires the ability to connect two separate nouns or noun phrases that refer to the same entity. The task gets especially complicated if the noun phrases in question have no string overlap at all, like *Obama* and *president* (compare *Barack Obama* and *Obama*, which is a relatively simple case)—a problem known as the ‘opaque mentions’ [3].

We regularly face this and other coreference-related issues in our ongoing work on named entity recognition (NER) and fact extraction. Relying on this experience, we are inclined to view coreference resolution as a subtask of entity recognition and identification in the broader sense of the word.

Even though the gold standard collection issued by the organizers did feature some examples of coreference between objects that could not be defined as named entities, these samples were relatively few. An overwhelming majority of coreferents tend to represent some kind of separate entity, either named or at least distinct and identifiable. Moreover, in most cases it was one of the ‘big three’ of NER—a person, a location or an organization. Therefore our approach mainly consisted in adjusting a set of ready-made entity extraction and identification rules to this particular task of coreference resolution. Nevertheless, some particular subtypes of coreference that could not be covered by the existing rules forced us to implement several new mechanisms, most notably a tool for graph-based semantic similarity measure that is described in the last section of this paper.

2.1. Candidate extraction

Two main stages of the process in our case are traditional for coreference resolution (see [2] for example), and include a) collecting all the probable candidates and b) filtering out those that do not seem to corefer with any other candidates. During the first stage we attempt to extract all the objects that could be identified as entities. Our entity extraction rules are generally based on the results of the ABBYY Compreno analysis and make use of the diverse linguistic information it provides (semantic classes, syntactic slots, semantic roles and many more, see [1] for details). The sets of rules vary for different types of entities. Here is a brief description of the core heuristics:

2.1.1. Person extraction

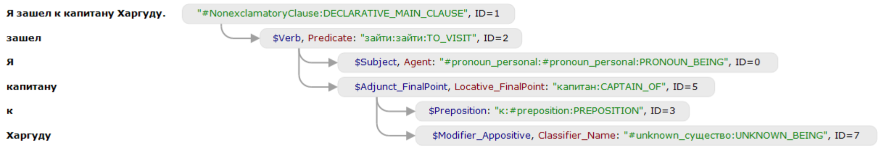
The task of person extraction in our system is subdivided into two major subtasks: detection of a person in a text and correct recognition of its attributes, i.e. name, surname, middle name and other parts of a proper name, if they are present. Extraction of attributes is essential for further identification of different textual instances as one person, as will be shown in the next sections.

The most obvious and straightforward way to locate a person in a text is by looking for a known personal proper name with capitalization. However, this simplistic approach alone rarely yields tolerable results, especially in terms of recall. First of all, even the most exhaustive databases cannot claim to have all the possible names and surnames, inevitably forcing a researcher to deal with the unknown ones. Secondly, there are many ambiguous names (*Bob, Virginia, Слава*), and even ones that lack ambiguity as such can still be used as proper names for entities other than human individuals (*пароход «Иван Федорович Крузенштерн», ресторан «Пушкин»*). Thirdly, a person can be referred to by a non-capitalized common noun/noun phrase (*мальчик, мужчина, космонавт, глава государства, state senator*).

The first problem—when a personal name is absent from the dictionary—can be addressed via syntactic and/or semantic structure. For instance, if a particular node of a parse tree has been labelled as an "UNKNOWN_BEING", we might try and look at the semantics of its parent. If the upper node turns out to be a name of a profession, a rank, an honorific or a nobiliary particle, chances are high that the node in question is a surname.

(6a) Input text
Я зашел к капитану Харгуду.

(6b) Semantic tree with syntactic slots



```
Я зашел к капитану Харгуду.  "#NonexclamatoryClause:DECLARATIVE_MAIN_CLAUSE", ID=1
зашел                          $Verb, Predicate: "зайти:зайти:TO_VISIT", ID=2
я                                  $Subject, Agent: "#pronoun_personal:#pronoun_personal:PRONOUN_BEING", ID=0
капитану                           $Adjunct_FinalPoint, Locative_FinalPoint: "капитан:CAPTAIN_OF", ID=5
к                                      $Preposition: "к:#preposition:PREPOSITION", ID=3
Харгуду                                $Modifier_Appositive, Classifier_Name: "#unknown_существо:UNKNOWN_BEING", ID=7
```

Fig. 8. Semantic tree with syntactic slots

Other personal markers include date of birth (*Гельмгольц, 1989 г.р.*), bracketed constructions with foreign words (*Кхиеу Порн (Khieu Porn) стал жертвой своих земляков*) or locations (*Вик Уайлд (Россия)—1 место*), certain verbs with strict selectional restrictions (*жениться, свататься*).

The issue of name ambiguity can be partially resolved by taking into account quotation marks and syntactic structure, which in some cases are determined by a particular meaning of an otherwise ambiguous proper name. Broader context might be helpful as well.

Addressing the third difficulty, when a referent of a person is a common name or a name phrase with no capitalization (*prime minister*), is particularly important for coreference resolution. However, at the stage of detection such cases pose little trouble—we basically mark any lexeme that fits semantically to define a person, is singular and complies with several other grammatical restrictions (so *prime minister* would fit and be extracted, as well as a *cosmonaut* or a *girl*).

As for the second subtask, the correct extraction of attributes (i.e. name parts) relies heavily on the common standards of writing down personal names. For instance, if we encounter a single initial followed by a capitalized word, the latter is usually a surname. Generally a complex personal name in our system is represented as a subtree with a surname or an initial as the top node. Its children might be a first name, several middle names or a patronymic, as well as initials or a part of a complex surname.

2.1.2. Organization & location extraction

The organization extraction rules fall into two main categories. Rules of the first category focus on keywords in the name of an organization itself and extract relatively straightforward mentions like *компания Тогрус* or *ОАО «Ромашка»* or *Cobham ltd*. They also deal with instances of enterprises and government bodies that are already known to the system by name.

Rules of the second category extract more obscurely-named organizations and rely on the context—mainly semantic classes and syntactic slots, but semantic roles are used sometimes as well. For instance, a rule that handles examples like *Он уволился из Омскэлектро* or *He resigned from RTRT* looks for a node with a semantic class "TO_RETIRE" and then creates an organization on its child provided that the latter has the semantic role of Locative_InitialPoint and is capitalized. Another rule that deals with corporate acquisitions (*Yahoo bought Tumblr*) requires a node with a semantic class "TO_ACQUIRE" with an Object in quotation marks among its children, while another child in the role of Possessor should not be a person (to exclude examples like *Vasya bought Sony Play Station*).

Proper names of the extracted organizations are stored as their 'identifier' attributes. Later on they are used at the identification stage.

Location extraction is based on the same principles. Keywords (*страна, город, озеро, bay, -city, creek* etc.) and sets of known proper names serve as the most reliable features, while previously unknown entities are derived with help of syntactic-semantic patterns. There are also additional stop-productions within the rules that do not allow the extraction of a known location in case it is used as a proper name for some other kind of named entities (*кафе Бомбей*).

The set of entities that are subject to extraction is not limited to these three types and includes a broad range of information objects from military aircraft to laws.

In these cases the general approach is quite similar to the one described above (while the exact properties of the extracted objects are, of course, different).

2.2. Identification and filtering

The first stage of the whole process can be described as a recall-oriented one, yielding a vast amount of referring expressions for further filtration. During the second stage the collected entities go through the identification process. The items identified as referring to one real-life object remain and form a coreference chain together, while the ones left without a pair are sifted out. This process determines the overall precision of the system, at the inevitable cost of decreasing the recall whenever an identification failure occurs. The identification rules rely chiefly on the attributes extracted during the entity extraction process. Following is a brief description of these rules for various entity types:

2.2.1. Person identification

The backbone of the identification of human-like entities is the intersection of attributes (name parts). For each pair of extracted persons the attributes are compared one by one, and if there is enough intersection and no contradictions, the objects can be merged. The discrepancy in gender prevents merging, so in case like *Иванов получил зарплату. Иванова рада* the entities will not be merged, whereas the two mentions of the same surname in *Иванов получил зарплату. Иванова обуяла радость* will be identified as relating to one person (this example demonstrated the advantages that complete syntactic-semantic analysis brings to coreference resolution).

Another way of person identification is via syntactic patterns combined with semantic restrictions. For instance, if a certain node with a person object attached to it has a nominal complement, we attach a special auxiliary link from the object to that complement. Then, if the same lexeme as in complement occurs elsewhere in the text, a second person is going to be extracted and the two person objects will merge due to that special link. Consider an example:

(7a) *Бьорндален—великий биатлонист. Спортсмен показал*
высший класс на олимпиаде в Сочи. Биатлониста такого
уровня нельзя списывать со счетов и после 40 лет.

In the first place our extraction rules locate three entities—*Бьорндален*, *биатлонист* and the second *биатлонист*. The two mentions of *биатлонист* are then merged into one person on the grounds of having similar semantic class, and after that the syntactic structure of the first sentence is used to identify *биатлонист* with the surname *Бьорндален*[1].

[1] Since the organizers of the contest chose not to consider coreference between a subject and its nominal complement, we did not connect them either. The described mechanism, nevertheless, was still used to identify and merge entities in the broader context. So in this particular case our coreference chain would show the connection between *Бьорндален* and *биатлонист* from the third sentence, but no visible link between the surname and the first *биатлонист* in the complement slot.

In order to extract the entire coreference chain from the last example one also has to identify биатлонист/Бьорндален with *спортсмен*. Fortunately, possession of an extensive semantic hierarchy allows us to do just that by incorporating certain WordNet-style graph-based metrics of semantic similarity into the identification process. In this particular case by traversing the hierarchical tree we find out that *спортсмен* is the direct hypernym of *биатлонист* and thus probably refers to the same person.

2.2.2. Organization and location identification

Organizations and locations are usually merged on the basis of their identifiers' (i.e. proper names) intersection. In addition to that there is a semantic similarity rule analogous to the one in person identification that was described above. Such a rule would merge *Роскосмос* and *контора* or *Роснефть* and компания in the following examples:

(8a) ***Роскосмос*** *запустил конкурента Google Maps. Государственная* ***контора*** *же, и деньгами налогоплательщиков работа оплачена.*

(9a) ***Роснефть*** *может получить контроль над всеми аэропортами Киргизии. Российская* **компания** *подписала меморандум о приобретении не менее 51 % ОАО «Международный аэропорт Манас».*

The identification will be possible because both *Роснефть* and *Роскосмос* are present in the semantic hierarchy and their semantic classes descend from these of the words *компания* and *контора*.

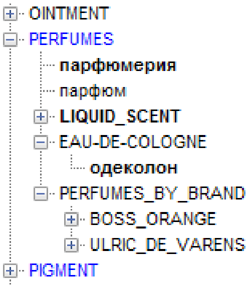
2.3. Adjustments for uncategorized entities

As has been mentioned before, the task of coreference resolution is not exactly limited to the identification of certain entities like individuals or organizations. In some cases coreferring expressions represent a real-life object that does not fall into any major entity category, and yet it is certainly supposed to be extracted.

A considerable share of such cases is constituted by demonstrative pronouns appearing as determiners (*лошадь—эта кляча; призрак—тот самый обозлившийся на него дух; аппарат—это устройство*). The resolution of this kind of coreference obviously requires some sort of semantic similarity data. As in case with common-noun persons, we use graph-based method. The idea behind this method is simple up-and-down tree traversal of the semantic hierarchy that yields synonyms as well as direct and indirect hypo/hypernyms. Whenever a demonstrative pronoun with a noun parent is encountered, the system launches a tree traversal procedure and the previous context is searched for a semantically similar noun. Here is an example from the test corpus of the competition:

(10a) *Я помню замечательный эпизод, когда она похвасталась нам с Володей Черняевым (он сейчас успешно работает в театре у Юрия Любимова) каким-то дорогим* ***одеколоном, который*** *она приобрела для молодого супруга. Мы попросили понюхать этот* ***парфюм.***

The semantic class of парфюм ("PERFUMES", which also includes *парфюмерия*) is the direct ancestor of the semantic class of *одеколон* ("EAU-DE-COLOGNE"), which enables us to unite the two objects. The relative pronoun *который* is replaced by its controller *одеколон* and attached to the coreference chain as well.



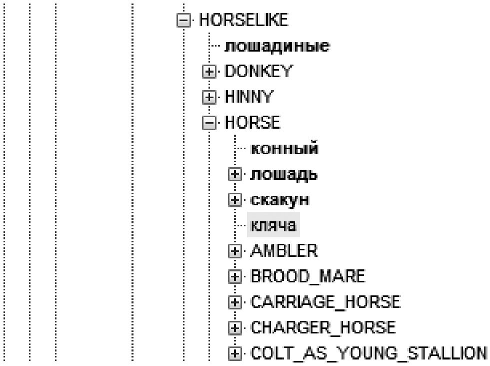
- ⊞ OINTMENT
- ⊟ PERFUMES
  - **парфюмерия**
  - парфюм
  - ⊞ **LIQUID_SCENT**
  - ⊟ EAU-DE-COLOGNE
    - **одеколон**
  - ⊟ PERFUMES_BY_BRAND
    - ⊞ BOSS_ORANGE
    - ⊞ ULRIC_DE_VARENS
- ⊞ PIGMENT

Fig. 9. A segment of the semantic hierarchy

Another example from the test corpus:

(11a) *Скоро ужасную* ***клячу****, словно сбежавшую с живодерни увидали*
и другие зрители. Люди смеялись, удивлялись, спрашивали,
негодовали. Как могла попасть сюда эта ***лошадь****?*

In this case two coreferents a) evidently represent an unnamed entity and b) are stylistic synonyms rather than hypo-hypernyms. In our semantic hierarchy the lexical classes *лошадь* and *кляча* exist within the same semantic class, and therefore the rule relying on demonstrative pronouns and semantic similarity applies to them as well.



- ⊟ HORSELIKE
  - **лошадиные**
  - ⊞ DONKEY
  - ⊞ HINNY
  - ⊟ HORSE
    - **конный**
    - ⊞ **лошадь**
    - ⊞ **скакун**
    - кляча
    - ⊞ AMBLER
    - ⊞ BROOD_MARE
    - ⊞ CARRIAGE_HORSE
    - ⊞ CHARGER_HORSE
    - ⊞ COLT_AS_YOUNG_STALLION

Fig. 10. A segment of the semantic hierarchy

Our experiments with the gold standard showed that this particular rule has very limited effect on the overall performance of the system, because the gain in recall is almost negated by the loss in precision, leaving F-measure increased by no more than a few per mille. But that can be explained by inconsistencies in the corpus markup (many legitimate cases of coreference with demonstratives were left unmarked by the contest organizers) and relative scarcity of such cases in the provided texts.

Unfortunately, our attempts to use this sort of semantic similarity methods on a broader scope did not prove successful, yielding too many false positive hits. However, it is acknowledged that most of the attempts to detect such 'opaque mentions' (i.e. with no string overlapping of nouns) tend to decrease precision significantly more than improve recall [3].

Another crude recall-oriented adjustment is simply the extraction of all the nodes with capitalized lexemes (except for those in the beginning of a sentence, of course) as well as lexemes and expressions in quotes. Each of them received two identifiers, a lemma of a given lexeme and the original word form that appeared in the text. Thus an information object *Нацбест* in *лауреат Нацбеста* has two identifiers—normalized *Нацбест* and original *Нацбеста*, which in one case helped us to identify two coreferents despite the normalization failure. At the identification stage such candidates were compared to each other and merged in cases of identifiers matching. Of course this adjustment is limited to unknown entities only and does not apply to persons or organizations.

Conclusion

Our approach to anaphora and coreference resolution has an obvious bias towards deep linguistic analysis (rather than the use of statistics and machine learning) and can be described as rule- or model-based. Such approaches are known to be relatively labour-intensive and have their limitations. However, the use of deep semantic data allows our system to perform well in many challenging cases like ambiguous examples of pronominal anaphora or 'opaque mentions' of coreferring expressions. Linguistic information also enables us to avoid such typical false positives as individuals with similar surnames but different gender.

We evaluated our system's anaphora resolution on a part of the training corpus. Since there were some inconsistencies in the gold standard, we double-checked all the discrepancies manually, so that the result was not lowered by the correct pairs detected by the system but absent from the training markup. This semi-automatic evaluation showed the F-measure of 0,644. We chose not to evaluate coreference resolution ourselves due to lack of agreement on evaluation metrics in this particular field (since whole chains are supposed to be evaluated rather than just pairs). It is expected that by the time this paper is published the organizers will have revealed the results of the independent evaluation.

References

1. *Anisimovich K. V., Druzhkin K. Ju., Minlos F. R., Petrova M. A., Selegey V. P., Zuev K. A.* (2012), Syntactic and semantic parser based on ABBYY Compreno linguistic technologies, Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog” [Komp’iuternaia Lingvistika i Intellektual’nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii “Dialog”], Bekasovo, pp. 90–103.
2. *Lee H., Peirsman Y., Chang A., Chambers N., Surdeanu M., Jurafsky D.* (2011), Stanford’s Multi-Pass Sieve Coreference Resolution System at the CoNLL-2011 Shared Task, Proceedings of the CoNLL-2011 Shared Task, Portland, Oregon, USA, pp. 28–34.
3. *Recasens M., Can M., Jurafsky D.* (2013). Same Referent, Different Words: Unsupervised Mining of Opaque Coreferent Mentions, Proceedings of NAACL-HLT 2013, Atlanta, Georgia, USA, pp. 897–906.
4. *Zuev K. A., Indenbom M. E., Judina M. V.* (2013), Statistical machine translation with linguistic language model, Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference “Dialog” [Komp’iuternaia Lingvistika i Intellektual’nye Tehnologii: Trudy Mezhdunarodnoj Konferentsii “Dialog”], Bekasovo, vol. 2, pp. 164–172.