ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Образовательная программа «Фундаментальная и компьютерная лингвистика»,
по направлению 45.03.03 Фундаментальная и прикладная лингвистика

Бутенко Зоя Алексеевна

RUSSIAN TIMEBANK

Выпускная квалификационная работа студентки 4-го курса

Академический руководитель	Научный руководитель
Доцент Школы лингвистики	Профессор Школы лингвистики
Ю.А.Ландер	О.Н.Ляшевская

Москва, 2023

Contents

Introduction

The main advantage of the human species' conscience is the ability to exist in not only the three physical dimensions, but also the fourth dimension – time. Unlike all other forms of life on Earth, we are able to model the world around us in the future, as well as analyze the past. Conserving, producing, and sharing that information with others is a great evolutionary advantage. The natural language being the main medium for exchanging information between humans has evolved to reflect our understanding of time, the complexity of which leaves great room for discussion among researchers in philosophy, linguistics, and many other fields of knowledge.

The instruments for conveying temporal information in language are naturally dependent on the peculiarities of time itself. For example, time is relative, which is why temporal expressions in a language tend to be anchored in something – usually, the moment of utterance:

(1) I went to the store ***today****.*

How do we know the exact moment in time when the speaker went to the store? We establish a relation between the anchor – moment of utterance, and the temporal expression – 'today'.

One might think that certain temporal expressions are absolute, for example, '14th August 1995' or '14:25'. However, even dates and times have a conventional anchor: the birth of Jesus Christ or midnight. Additionally, temporal expressions and other indications of time in a text do not refer to a specific moment in time, but rather an interval, which makes the relationships between them even more complex. Since time measurement is conventional and arbitrary, just like any physical measurement – a meter, a kilogram and so on – it can always be broken down into smaller elements. Therefore, any moment in time is also an interval, which can be related to other intervals in many different ways, including overlap, precedence etc. Parsing temporal information in a text is not only extracting indicators, but also anchoring them in time and establishing the relationships between them.

The way human languages conceptualize time is a very important cognitive problem. The famous Time is Space metaphor, first described in (Lakoff 1980) explains many universal representations of time in languages. Mapping the way we see the three physical dimensions onto the fourth one seems to be reasonable since they do share some key characteristics. For example, just like one can move forward in space, one can move forward in time. We can describe the positioning of physical objects by defining which of them is closer to us just like how we describe which of the events happened before the other. In a way, both time intervals and physical locations require an anchor to be properly described.

Reconstructing relationships between time intervals is not a trivial task. Time intervals are expressed by a set of linguistic instruments: temporal indicators (‘tomorrow’), modifiers (‘ago’ as in ‘a year ago’) and quantifiers (‘two’ as in ‘two summers from now’). In addition, anaphoric contexts, metaphors, etc. are obstacles in reconstructing a timeline of events. Therefore, automatic extraction of temporal information from a text is an incredibly important and difficult task. As a part of a bigger scope of tasks called Information Extraction (IE) it is aimed at automatically extracting relevant information from various sources. Developing a system for automatic parsing of temporal information will allow us to construct a temporally annotated corpus, which would have a variety of applications in both computational and theoretical linguistics.

Many attempts have been made to create such a resource for other languages, the most famous one being the TimeBank 1.2 (Pustejovsky 2003). Other TimeBanks have followed the same annotation standards, some have made necessary modifications. There have not yet been any attempts at creating a similar resource for Russian, as well as other Slavic languages. Additionally, the annotation scheme initially designed for English (though with a universal application in mind) has not yet been adapted to Russian.

Creating a TimeBank for Russian implies the following stages of work:

1. Tailoring the universal annotation schema to the specifics of the Russian language;
2. Gathering and preprocessing data;
3. Annotating a gold standard for evaluation;
4. Creating various pipelines for automatic annotation of texts with temporal information;
5. Evaluating different approaches and annotating the corpus with the optimal pipeline.

Thus, the goal of this work is to tailor the universal annotation schema to Russian, try different approaches to temporal and event annotation, evaluate different pipelines in order to find the optimal approach, and produce a temporally annotated corpus.

The paper consists of the introduction, six chapters, results, conclusion, and references.

Chapter 1 provides theoretical background on the category of time in Russian, as well as other grammatical categories that are relevant to the task of information extraction in Russian.

In the second part I review previous attempts at building similar resources, their data and methodologies.

In the third part we explain the data used for the creation of the resource, as well as the preprocessing required.

In the fourth part we explain the methodology for building the Russian TimeBank, including the methods used for Event Extraction, Time Stamping, establishing relations and links between temporal expressions and event instances.

In the fifth part we review the results of the evaluation of different approaches.

Finally, in the sixth part we discuss challenges associated with the work and prospects for future research.

Chapter I. Theory

1.1. Temporality

The category of time is a complex grammatical and semantic phenomenon. Temporal information is necessary to convey the time an event takes place, which is why events and temporal expressions are closely intertwined. The concept of time is usually understood as neutral or objective, however, the way humans experience time can never truly be objective. Moreover, we now know that time can slow down or speed up from the perspective of physics, for example, depending on the gravitational pull. Since time is entirely subjective, a more accurate term for time for our purposes is *temporality*, or the way time is experienced by an individual. In this paper temporality is understood as the way events are placed on the time continuum in relation to each other.

Any event or occurrence, even seemingly instantaneous, takes up a duration of time. It is thus more appropriate to conceptualize events as time intervals rather than moments in time. This allows us to better understand the relationships between them. James F. Allen was the first to introduce interval algebra (Allen 1981), which has since been used as the basis for temporal description of events. According to Allen, there are 13 base relations between intervals, including precedence, overlapping and others.

The 13 base relations between intervals are the following:

Relation	**Inverse relation**	**Chronological Sequence**
X before Y	Y after X	Xstart < Xend < Ystart < Yend

X equals Y	Y equals X	Xstart = Ystart < Xend = Yend
X meets Y	Y met by X	Xstart < Xend = Ystart < Yend
X overlaps Y	Y overlapped by X	Xstart < Ystart < Xend < Yend
X contains Y	Y during X	Xstart = Y start < Yend < Xend
X starts Y	Y started by X	Xstart = Ystart < Xend < Yend
X finishes Y	Y finished by X	Ystart < Xstart < Xend = Yend

Table 1. 13 base intervals according to Allen's interval algebra

The base relations of intervals in Allen's interval algebra are used to link event instances as well as temporal expressions in the universal annotation standard TimeML.

Since temporality implies a link between time intervals and events, the second stage of TimeBank annotation is event extraction, one of the most important steps of which is correctly defining an event. It is most often defined as 'anything that happens'. However, conventionally events are understood as an occurrence of somewhat extraordinary significance. The classification becomes even more complicated when we consider that events can also be represented by states which have a start and an end. For example:

(2) Last winter, my grandfather ***fell sick****.*

(3) My grandfather ***is sick****.*

(4) My cat ***is funny****.*

In example (2) the event has a clear starting point. Example (3) is arguably an instance of the same event extending into the future. The example (4) would then also be considered an event, since it could have a start and an end (even if the cat has been funny its entire life).

The annotation standard used for the original TimeBank – TimeML – defines events as 'situations that happen or occur', as well as 'predicates describing states or circumstances in which something obtains or holds true' (Saurí 2006). Although the definition is incredibly vague and leaves a lot of room for interpretation, our work is aimed at developing a resource based on the universal annotation standard TimeML. The annotation is thus dependent on the understanding of the individual annotator, which in turn poses certain challenges for both developing the gold standard as well as evaluating the pipeline for automatic event extraction. The accuracy of the automatic annotation is not easily measurable in this task, since in a lot of cases multiple interpretations are possible.

The way other TimeBanks approach this issue is annotation in parallel. The standard pipeline for manual annotation is two annotators working in parallel and a third annotator resolving any conflict. However, in our case it is not possible to involve multiple professionals, and it would only solve the issue when it comes to the gold standard. Although the automatic extraction algorithm can be trained on the gold standard, it would 1) reflect the initial annotators' view of what an event is; 2) require a lot of manually tagged data.

1.2. Morphology and Syntax

Across languages, temporality is expressed in a great variety of ways, both lexically and grammatically. Although time is relatively universal across human languages, temporality is not: different languages express aspect, mood etc. using different morphosyntactic instruments.

One of the obvious differences between English and Russian in terms of temporality is aspect. Every Russian verb can be assigned an aspect value – perfective or imperfective – depending on some explicit device (prefix, suffix, stem alteration etc.) signaling aspect. Most simplex verbs (i.e. without a prefix as the signaling device) are imperfective, though there is a small subset of simplex verbs that are perfective and some that are ambiguous. Adding a prefix to a simplex verb usually results in a perfective verb form, the prefix is then 'empty' in meaning and only adds a manner of action meaning (for example, makes the verbs inchoative). The imperfective meaning is usually formed by adding a suffix to the simplex verb form. (Klein 1995)

The diversity of morphological instruments used to change the aspectual meaning of the verb in Russian leads to a plethora of fascinating phenomena, such as, for example, secondary imperfectives. An imperfective suffix can be attached to a prefixed perfective verb, rendering it imperfective. The lexical meaning of the verb thus stays the same, and only the aspectual meaning is changed. E.g.:

(5)	li-t' (IMP1)	raz-li-t' (PERF)	raz-li-va-t' (IMP2)
	pour-INF	PR-pour-INF	PR-pour-IMP2-INF
	'to pour',	'to pour out',	'to pour out', 'to spill',
	'to be pouring'	'to spill'	'to be pouring out'

(Galambos 2007).

There have been many attempts to describe the semantics of the perfective and imperfective aspect in Russian, though there is not a generally accepted definition. The main semantic differentiation is that perfective verbs imply a certain boundary, and imperfective verbs seemingly do not. There are then many submeanings of perfective and imperfective aspects, which all share this characteristic. For example, the imperfective aspect sometimes refers to habitual actions, as in 'We go to the cinema on Fridays' – there are no clear temporal boundaries for the action.

Unlike Russian, English does not form aspectual meanings using affixes. Instead, auxiliary verbs are used, for example the verb 'to be' in the progressive aspect ('I was studying') and 'to have' in the perfective aspect ('I have done my homework'). This important distinction affects not only the way aspect is parsed automatically, but also how aspect is annotated. Aspect is one of the most important attributes in the annotation standard, since it indicates the time interval (or a set of intervals) during which the event takes place.

Another important grammatical category that differs between languages is mood. The annotation for English is quite straight-forward: mood is usually represented by a modal auxiliary which can simply be tagged in one of the event instance attributes. In Russian, however, modal expression is more diverse. For example, the subjunctive mood is rarely expressed in contemporary English texts, but is very frequent in Russian. It can be used to express politeness or as a form of indirect speech. The variety of morphosyntactic devices is also greater in Russian: the category of mood is not only expressed by the auxiliary verbs. For example, the conditional mood is formed by adding a *бы* particle.

(6) *She* ***would go*** *with you if she weren't busy.*

(7) *Она* ***бы пошла*** *с тобой, если бы не была занята.*

Although both Russian and English employ past, present, and future tenses, there are morphological differences when it comes to tense as well. Morphologically, the Russian tense system distinguishes past and non-past tenses, not the typical triad of past, present, and future. The main factor that can complicate interpretation of tenses is the complex Russian aspect system described above. However, most automatic parsers and taggers can easily distinguish between tenses. Knowing the time of utterance (in our case, the date of the document creation) allows us to easily place an event on the timeline before, after or during the time of utterance, depending on the tense.

Other than the TAM system, there are several other morphological categories in Russian that could be relevant in our work. For example, gender can be used for disambiguation in anaphoric contexts:

(8) *В августе мы поедем на лингвистический съезд и на* ***конференцию***. ***Она*** *пройдет в двадцатых числах.*

Since events can be represented by nouns, it is important to consider their gender and number when annotating Russian texts. This is something that has not been done in the original TimeML annotation standard, since it is irrelevant for English.

Similarly, Russian noun cases can provide relevant information, specifically when it comes to temporal expressions.

1.3. Temporal Expressions

Although the basic set of temporal expressions is relatively universal – concepts such as *day, week, year, second, month* etc. – there are significant differences in the modifiers and quantifiers of temporal expressions. For example, 'this week' in English does not require modifiers, whereas in Russian 'на этой неделе' employs both a preposition and a specific case to convey the same information. The collocations of prepositions and nouns used for most temporal expressions can be easily extracted by studying bigrams in Russian text, however, the relationship between temporal expressions and grammatical cases is far more complex. Luckily, case is also an important indicator that may help understand the extendedness of an event. For example, 'новый год' can be understood as both the celebration of the new year's eve and the entirety of the upcoming year. However, the case used helps resolve the polysemy: 'в новом году' – 'in the upcoming year', 'в новый год' – 'during new year's eve' (Nesset 2004).

The set of temporal expressions in a language can be conceptualized as a relationship between a set of primitives (*day, month, second, etc.*), their modifiers and quantifiers. For example, the temporal expression 'twice a week' represents a set of events as indicated by the quantifier 'twice', while 'during the week' is an interval as indicated by the modifier 'during'. The task of extracting temporal information is thus often solved using a rule-based approach, where a predefined set of modifiers, a predefined set of quantifiers, and a predefined set of temporal markers are used to extract the semantics of the temporal expression as a whole.

Temporal expressions then interact with the value of the anchor modifying it in a certain way. For example, if the time of utterance is 1st of January 2003, the temporal expression this week refers to an interval from 30/12/2002 until 5/01/2003. The process of inferring the boundaries of a time interval from a temporal expression is called normalization.

Chapter II. Typology of TimeBanks

There is a variety of TimeBanks for different languages, including English, Spanish, Catalan, French, German, etc. All TimeBanks are annotated using a universal standard for temporal annotation – TimeML, allowing to study and compare temporality across the aforementioned languages. TimeBanks share a lot of similarities in data types, volumes of data, parts of methodology, etc. However, the morphosyntactic properties of languages often require some type of adaptation to both the annotation standard and methodology.

2.1. TimeML

The first TimeBank was introduced in 2003 (Pustejovsky 2003) as an illustration corpus for the new universal annotation standard for temporal information – TimeML (Ingria, Pustejovsky 2002). The annotation standard was based on the Extensible Markup Language (XML), which is a simple text format for structured information. The annotation standard was developed to address the following tasks:

1. Time stamping of events (identifying an event and anchoring it in time);
2. Ordering events with respect to one another (lexical versus discourse properties of ordering);
3. Reasoning with contextually underspecified temporal expressions (temporal functions such as 'last week' and 'two weeks before');

4. Reasoning about the persistence of events (how long does an event or the outcome of an event last).

(URL: https://timeml.github.io/site/index.html)

The following tags are used in the annotation scheme:

Tag	**Description**
TIMEX3	This tag is used to capture dates, times, durations, and sets of dates and times.
EVENT	The tag is used to annotate elements in a text that mark the semantic events described by it.
MAKEINSTANCE	A realization link; it indicates different instances of a given event.
SIGNAL	The tag is used to annotate temporal function words such as "after", "during", and "when". These signals are then used in the representation of a temporal relationship.
TLINK	Temporal links are represented with a TLINK tag. A TLINK can temporally relate two temporal expressions, two event instances, or a temporal expression and an event instance.
SLINK	This tag is used to capture subordination relationships that involve event modality, evidentiality, and factuality.
ALINK	An aspectual connection between two event instances is represented with ALINK.

Table 2. Description of tags used in the TimeML annotation standard

Each of the tags has a set of attributes, for example, the MAKEINSTANCE tag has a non-optional attribute ‘pos’, which indicates the part of speech of the word that reflects the event instance. The set of tags and sets of attributes for each tag are predefined, and although the annotation standard was designed to be universal, many attempts at applying it to other languages have led to the creation of new annotation variants. For example, the Italian TimeBank (Caselli 2011) has introduced a separate version of TimeML tailored to the morphosyntactic specifics of Italian – It-TimeML.

Firstly, new attributes were added to the EVENT tag, one of which is MOOD. The decision to add such an attribute directly to the EVENT was based on the fact that the category of mood in Italian is reflected as an affix on the lexical verb, not as an auxiliary like in English. Another added attribute was VFORM, which is used for differentiation between finite and non-finite forms.

A very important distinction between Italian and English is aspect, which is why two new values for the aspect attribute were added: PERFECTIVE_PROGRESSIVE and IMPERFECTIVE. Overall, many of these changes are also relevant for the annotation scheme for Russian, including the MOOD attribute and the values for the aspect attribute.

The It-TimeML annotation standard was not only used for the Italian TimeBank, it has also been adapted for the German TimeBank KRAUTS – Korpus of newspapeR Articles with Underlined Temporal expressionS (Strötgen 2018). One of the reasons It-TimeML was chosen as the basis for the annotation of the German TimeBank is morphological similarities, such as the fusion of prepositions and articles (*im* = *in* + *dem* 'in the').

In addition, German morphology is rich in compounds where parts of the compound act as a temporal expression, unlike English, where the boundaries of temporal expressions usually coincide with the boundaries of a word or a phrase. For example, *Diskussionabende* '*Evenings of discussion*'. Only compounds where the temporal expression is the syntactic head of the compound can be considered temporal expressions, which is why *Werktag* 'work day' must be tagged as a temporal expression, whereas *Monatsblatt* 'weekly leaflet' must not.

Other TimeBanks such as Spanish (Saurí 2012), Catalan (Saurí 2012), and French (Bittar 2011) also use TimeML with adaptations to the language's morphology. Spanish and Catalan TimeBanks incorporate changes into the mood and aspect attributes, while the French TimeBank suggests a new value for the aspect attribute – PROSPECTIVE (for *aller* + Vinf 'to be going to + Vinf').

2.2. Data

Temporal annotation of texts, specifically a creation of a TimeBank, requires an initial corpus with a high content of both events and temporal expressions. Texts which are strictly descriptive are not very valuable, since there are virtually no temporal anchors or links. For example, corpora of social media texts are unlikely to contain many event instances or temporal anchors, moreover, the short length of a median social media post suggests there are often no links between events. It appears that long and complex narratives are the most valuable in this case, however, most fictional texts have long dialogues, descriptions, and author's monologues, which are not rich in events. In addition, fictional texts are rarely anchored in the date of creation, while reconstructing the anchor time is often impossible. The best choice of data for creation of a TimeBank seems to be news corpora, which are rich in events and usually include explicit mentions of dates, times, and intervals.

The original TimeBank (Pustejovsky 2003) uses 132 Wall Street Journal documents (~61000 tokens in the final dataset), the German KRAUTS uses texts from Dolomiten and Die Zeit newspapers (75678 tokens in the final dataset), the French TimeBank uses Est Républicain corpus of journalistic texts (16208 tokens in the final dataset) etc. While it is a noble goal to create a pipeline for reconstructing the timeline in fictional narratives, news corpora are better fit for the task of event extraction as well as annotation of temporal information.

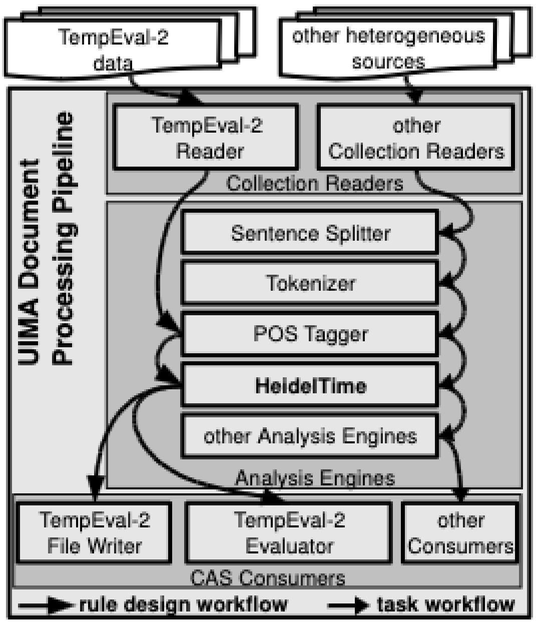
2.3. Methodology

Temporal annotation of a corpus consists of several tasks: event extraction, time stamping, linking event and temporal extractions etc. There are many different ways to approach every one of those tasks, so it is important to look at the methodologies of previous approaches at building a TimeBank. Although TimeBanks are not usually created to build a pipeline of automatic annotation, most of the time they do employ automation at some of the steps.

TimeBank 1.2 uses Stanford's bidirectional dependency network tagger to get POS-tags, which were then used to identify finite verb forms. The following annotation is carried out using the GraphAnno Python library[1] which creates graph representations for each clause and makes it possible to create links between nodes. For example, the automatically extracted finite verb forms are linked to the TOPICTIME node with aspect. This is based on the assumption that aspect is the relationship between the topic time and the time of situation (Klein 1994). The annotation was inferential, meaning a manually annotated portion was used to infer new annotations. (Gast 2016)

Overall, the annotations are partly manual, partly inferential, and partly automatic. This is not the most convenient pipeline, since a portion of the corpus has to be pre-annotated and the automatically retrieved information still needs to be corrected and supplemented manually.

Another approach is the HeidelTime system. The system is strictly rule-based, mainly using regular expression patterns while scoring great results (86% F-score in TempEval-2).



[1] https://pypi.org/project/graphanno/

Figure 1. HeidelTime pipeline

The pipeline is a UIMA (Unstructured Information Management Architecture) component which consists of three main steps: a Collection Reader which accesses the documents and initializes a CAS (Common Analysis Structure) object, the analysis performed by the Analysis Engines, and CAS Consumers for final processing. (Strötgen 2010)

The HeidelTime system has been manually developed for 13 languages as of now, including Russian, and has automatically crafted parsers for more than 200 languages. However, the demo-version[2] for Russian does not show adequate results. For example, here is the result for one of the texts in our gold standard:

```
<?xml version="1.0"?>

<!DOCTYPE TimeML SYSTEM "TimeML.dtd">

<TimeML>

Итальянский премьер-министр Сильвио Берлускони готовит материал для нового альбома любовных песен. Об этом сообщает AFP. По данным агентства, пластинка выйдет уже в <TIMEX3 tid="t1" type="DATE" value="2010">2010 году</TIMEX3>.Готовы уже восемь из 12 песен, которые должны войти в альбом. Однако название пластинки и ее обложка пока держатся в секрете.Новый альбом будет записан политиком в соавторстве с гитаристом Мариано Апичелли. Вместе с ним Берлускони выпустил уже четыре пластинки. В <TIMEX3 tid="t4" type="DATE" value="2003">2003 году</TIMEX3> Апичелли и Берлускони даже написали песню, посвященную Владимиру Путину.В последние <TIMEX3 tid="t9" type="DURATION" value="P1W">недели</TIMEX3> Сильвио Берлускони поправляется после пережитого в <TIMEX3 tid="t8" type="DATE" value="2023-12" mod="START">начале декабря</TIMEX3> нападения. Психически неуравновешенный житель Милана швырнул в политика тяжелую сувенирную статуэтку. У премьер-министра были выбиты два зуба и сломан нос.

</TimeML>
```

Although rule-based systems show the best results for normalization of temporal information, machine-learning approaches are now superior in the extraction of temporal information. As we can see from the example above, the model easily normalizes dates and times, but struggles to extract all relevant temporal information. For example, the temporal expression ‘в последние недели’ was

[2] https://heideltime.ifi.uni-heidelberg.de/heideltime/

extracted only partially as ‘недели’ and assigned a time interval of one week instead of several weeks prior to the document creation time.

An alternative is combining the rule-based approach with machine-learning. The TRIPS and TRIOS system is a combination of a rule-based deep semantic analysis system (TRIPS) and a Conditional Random Fields classifier (TRIOS). The main advantage of the system is the ability to use it on raw texts as opposed to a corpus pre-tagged for events, like many other systems. (UzZaman, Allen 2010)

Chapter III. Data

As mentioned above, news reports corpora are optimal for extracting events and temporal information. Long and complex narratives in fictional texts have limitations in terms of normalization of temporal information, as well as anchoring events in time, reducing the text to a reconstruction of the timeline of events, rather than fully annotating temporality. Corpora of oral speech or other informal texts such as social media posts have specific limitations, such as insufficient length. The initial corpus of texts should therefore mostly consist of news reports which are rich in events and explicit temporal indicators.

The Taiga corpus[3] is an open-source corpus divided into several subcorpora, including the news subcorpus (Shavrina, Shapovalova 2017). It includes texts from Lenta.ru, Interfax, KP, Fontanka, and others. The entire corpus contains 92 million tokens, however, for our task we only need a section comparable to other similar resources (~60000 tokens). Although each text is provided with a pre-tagged annotation for part of speech as well as Universal Dependencies, our goal is to process and annotate raw texts.

[3] https://tatianashavrina.github.io/taiga_site/

An important criteria for the corpus is the availability of the main temporal anchor – document creation time (DCT). The Taiga News subcorpus stores the information about the date of creation of each text in the corpus, which makes it possible to anchor all temporal expressions in the text.

The final dataset includes 82 processed texts chosen randomly from the Taiga News subcorpus.

Creating an optimal pipeline requires testing different models and algorithms at each of the stages, the results of which need to be evaluated in order to choose optimal approaches at each step of processing and annotating. This requires a gold standard, which has to be annotated manually and be comparable in size to the test and validation datasets for TempEval (~6000 words). All texts need to be represented by a valid XML file annotated in accordance with the universal TimeML standard.

Chapter IV. Methodology

4.1. Event Extraction

Event Extraction (EE) is a sub-task of Information Extraction (IE) which usually includes identifying an event trigger (the word most explicitly signaling an event), event arguments (actors) and temporal information that can be linked to the event instance. Event Extraction is the first and one of the most important stages of building a temporally annotated corpus (a TimeBank). The arguments of events are irrelevant in terms of annotation of the TimeBank, the main focus being the event instances, temporal information, and the relations between them. EE can be approached in many different ways, the most popular now being machine-learning. Most machine-learning approaches in IE are supervised, which requires a fairly large training dataset. Not only would it be very time-consuming to annotate a big portion of the corpus manually, additionally, we have to consider the ambiguity of event classification: the definition of an event is incredibly vague, especially for Russian, which does not have many resources annotated for events or temporal information.

The EE algorithm should therefore ideally be unsupervised or at the very least require minimal volume of pre-annotated data. A great advantage of news reports texts is the fact that the majority of events are expressed by prototypical devices, mainly finite verbs.

The prototypical event trigger is a finite verb, usually in past tense or present tense. Although events can also often be represented by nominalizations, pronouns, adjectives, etc., the majority of events are prototypical when it comes to news reports. Therefore, the first step to event extraction has to be POS-tagging and dependency parsing, the annotation for which can be used to extract finite verbs in past or present tense that act as the predicate of the clause. Additionally, verbs that express events can act as clausal modifiers. For example:

(9) ***Находившиеся*** *в машине люди начали стрелять в сотрудников правоохранительных органов и были уничтожены ответным огнем.*

We will use the current standard for syntactic annotation – Universal Dependencies[4] (UD) – for all morphological and syntactic information. According to the UD standard of annotation, the event trigger in (9) is considered an adnominal clause modifier, as it modifies the nominal *люди*.

Event mentions can also act as adverbial clause modifiers (10), which is a clause that modifies another clause as a modifier, not a core complement.

(10) *Исчезновение картины с выставки было замечено утром в четверг, когда экспозиция* ***открылась*** *для посетителей*.

Although news texts usually contain finite verbs that are predominantly event instances, there are still a number of verbs that never appear in such contexts (e.g. *нравиться, являться,* etc.). Semantic frames can be used to filter out such verbs based on the semantic classification of verbs.

Aside from semantic information contained within the verb itself, there are other instruments that can signal that the verb is not used in the context of an event instance, including modality. This is especially important when dealing with verbs in present tense and can filter out contexts such as (11).

(11) *Он может* ***быть занят***.

(12) *Он* ***занят***.

Although we would consider (12) a stative event that occupies a time interval, (11) cannot be annotated as an event.

[4] https://universaldependencies.org/introduction.html

Although finite verbs are fairly easy to extract and most often express events in news texts, there are other parts of speech that are used to express events for which it is much harder to determine whether they express events in a text. We therefore need to find ways to extract less prototypical events.

An obvious advantage of a news corpus is the fact that event arguments are often named entities which are fairly easy to extract automatically. A lot of EE algorithms rely on event arguments, which can be used as a clue that the word connecting them syntactically has semantics of an event. The task of Event Extraction can then be reduced to Named Entity Recognition (NER) + Dependency Parsing + resolving semantic relations. The Shortest Path Between Entities approach described in (Rusu et al. 2014) consists of the aforementioned steps and finding the shortest paths between Named Entities. The events are then disambiguated using semantic resources such as WordNet, FrameNet, or others. Moreover, clustering algorithms can be used to classify events into states, occurrences, reporting, etc.

Such an algorithm requires testing different POS-taggers, dependency parsers, and Named Entities Recognizers. Testing different models for each stage of the pipeline and evaluating the results on the gold standard will allow for the selection of the optimal pipeline for event extraction.

First, named entities are extracted from the texts. There are a variety of models developed for Russian that use different instruments and algorithms to extract named entities. For our purposes, three models will be tested: Natasha, Stanza, and SpaCy.

Natasha is a set of libraries for Russian NLP that solves a number of tasks, including NER, tokenization, lemmatization, POS-tagging, and syntax parsing. The model is quite compact and does not require a GPU, unlike DeepPavlop which performs slightly better on NER (1% difference in F1-score on factRuEval-2016 (Starostin et al. 2016)). The model performs best on news texts which is an advantage for our data, in addition, the model was designed and trained specifically for Russian. The authors' evaluation on four datasets (factru, gareev, ne5 and bsnlp) also suggests better performance than SpaCy and Stanza[5]:

	factru			gareev		ne5			bsnlp		
	PER	LOC	ORG	PER	ORG	PER	LOC	ORG	PER	LOC	ORG
stanza	0.943	0.865	0.687	0.953	0.827	0.923	0.753	0.734	0.938	**0.838**	**0.724**
spacy	0.901	0.886	0.765	0.970	0.883	0.967	0.928	0.918	0.919	0.823	0.693
slovnet	**0.959**	**0.915**	**0.825**	**0.977**	**0.899**	**0.984**	**0.973**	**0.951**	**0.944**	0.834	0.718

Table 2. Comparison of F1 scores for stanza, spacy, and slovnet (base model for natasha's NER).

Spacy is another set of libraries for basic NLP tasks that provides support for 72+ languages with a number of pre-trained models. The model selected for our pipeline is 'ru_core_news_lg', which is pre-trained on a rather large corpus of news texts. It claims to provide a 0.95 accuracy on NER as well as dependency parsing.

Stanza is a multilingual Python NLP-toolkit that has a fully neural pipeline and produces annotation for tokenization, morphology, dependency parsing, NER, etc. The estimated F1-score of NER for Russian is 0.926 (Peng Qi et al. 2020).

We will use a number of models and pipelines for both POS-tagging and dependency parsing in order to extract not only finite verbs, but also various types of clause modifiers that can express an event. They include the aforementioned Stanza, SpaCy, and Natasha.

The embedding model for Natasha (Naeval) provides evaluation comparisons for five datasets (news, wiki, fiction, social, and poetry) from GramEval2020

[5] https://github.com/natasha/slovnet#ner-1

(Lyashevskaya et al. 2020). The evaluation below are F1-scores for Stanza, SpaCy, and Slovnet for each of the five datasets[6].

	news	wiki	fiction	social	poetry
stanza	0.934	0.831	0.940	**0.873**	**0.825**
spacy	**0.964**	**0.849**	**0.942**	0.857	0.784
slovnet	0.961	0.815	0.905	0.807	0.664

Table 3. F1-scores for Stanza, SpaCy, and Slovnet for morphological features.

The models are also compared in terms of time and memory efficiency (initiation time, disk space allocated, RAM space allocated, iterations per second).

	init, s	disk, mb	ram, mb	speed, it/s
stanza	2.0	591	393	92.0
spacy	8.0	140	579	50.0
slovnet	**1.0**	**27**	**115**	**532.0**

Table 4. Time and memory efficiency for Stanza, SpaCy, and Slovnet (morphological parsers).

The same evaluations are available for syntactic parsers of the aforementioned models[7]:

	news		wiki		fiction		social		poetry	
	uas	las	uas	las	uas	las	uas	las	uas	las
stanza	0.940	0.886	0.815	0.716	**0.936**	**0.895**	0.802	0.714	**0.713**	0.613
spacy	**0.943**	**0.916**	**0.851**	**0.783**	0.901	0.874	**0.804**	**0.737**	0.704	**0.616**
slovnet	0.907	0.880	0.775	0.718	0.806	0.776	0.726	0.656	0.542	0.469

Table 5. F1-scores for Stanza, SpaCy, and Slovnet for syntactic information.

Comparisons of speed and memory allocation:

	init, s	disk, mb	ram, mb	speed, it/s

[6] https://github.com/natasha/naeval#morphology-taggers
[7] https://github.com/natasha/naeval#syntax-parser

stanza	3.0	591	890	12.0
spacy	9.0	140	579	41.0
slovnet	**1.0**	**27**	**125**	**450.0**

Table 5. Time and memory efficiency for Stanza, SpaCy, and Slovnet (syntactic parsers).

As seen from tables above, the three models perform differently on each of the tasks (NER, morphology, syntax parsing). Natasha shows the best F1-scores on NER, while Stanza and SpaCy perform better when it comes to morphology and syntax parsing. Nonetheless, time and memory efficiency is also important for fairly large corpora. Natasha outperforms other models on both morphological and syntactic parsing.

The morphological and syntactic information is not only useful for event extraction, but also for further processing of events (annotating morphological attributes for event instances, extraction of links, etc.).

In addition, all events have a non-optional attribute ‘class’ in the TimeML standard. The set of values for the attribute is: OCCURRENCE, PERCEPTION, REPORTING, ASPECTUAL, STATE, I_STATE, I_ACTION. The attribute is intended to provide semantic classification of events, which is used by some of the LINK tags. Annotation of the semantic class of an event can either be done using machine-learning (which would require a training dataset) or using an existing database of semantic classes of verbs. The challenges associated with the annotation for the ‘class’ attribute are discussed in Chapter VI and leave the tag annotation out of the scope of this paper.

The next stage of event extraction is establishing event instances. According to the TimeML standard, a MAKEINSTANCE tag has to be created for each instance or realization of an event. This is necessary for the cases where multiple events are expressed once explicitly. For example:

(13) *На этой неделе он* ***сходил*** *в кино в понедельник и в четверг.*

(14) *Вчера я дважды* ***позвонил*** *маме.*

Additionally, the MAKEINSTANCE tag carries morphological information about event instances in its attributes, such as part of speech, tense, aspect, modality, polarity. Each of the attributes can be annotated using syntactic and morphological parsers described above.

Since an event can have multiple instances in a text, a lot of times there is only one explicit event mention, whereas other instances are either implicit or refer to the first event mention. In the latter case, the task of establishing event instances is anaphora or coreference resolution.

(15) *"Даже более мягкие вещи сегодня уже не проходят. Потому что те, кто не заинтересован в их прохождении, их* ***блокируют****. И делают* ***это*** *эффективно", —* *заявил глава государства.*

Most models and instruments developed for anaphora and coreference resolution are suited for nominal phrases only since the aim of both anaphora and coreference resolution is to find all references to the same real-world entity [Ju et al. 2014], whereas event instances refer to an occurrence or process. It is thus especially challenging to classify whether two event mentions are instances of the same event or not, even when it comes to manual annotation.

Considering the lack of pre-trained models and the challenges associated with resolving anaphora and coreference for event instances, this task is out of the scope of this work.

4.2. Extraction of Temporal Expressions

Rule-based approaches described in 2.3 are famously great at normalization of temporal information. Although there are now many machine-learning approaches that perform better on the extraction of temporal information than rule-based systems, most of them are supervised and require a big amount of training data.

Extracting temporal indicators and their modifiers can generally be easily approached with a set of rules and regular expressions, especially when there is not enough training data available (as in the case of Russian).

We will use the HeidelTime system [Strötgen et al. 2010] for the extraction of temporal indicators (TIMEX3 tag) and their modifiers. The system offers annotation for both news texts and narratives.

The HeidelTime system utilizes TreeTagger for POS-tagging and lemmatization. TreeTagger is a probabilistic model where probabilities are estimated using a decision tree [Schmid 2013]. We use the standalone version of HeidelTime [Zell, Strötgen 2015].

Running the HeidelTime standalone on raw text files with DCT yields a TimeML annotation for temporal expressions. Each temporal expression is wrapped in a TIMEX3 tag and normalized.

Although the extraction as well as normalization of temporal expressions is accurate most of the time, the HeidelTime system does not extract relations between temporal expressions which is one of the most important steps in temporal annotation. Additionally, HeidelTime does not extract signals (for example, *в* as in *в четверг*) which are used to link event instances to temporal information. Establishing links and relations between elements of annotation is therefore a task that needs to be solved separately.

4.3. Links and Relations

The TimeML annotation standard uses three tags for links: ALINK, SLINK, and TLINK. ALINKs are used for aspectual connections between event instances, for example, initiation of one event instance by another as in (16).

(16) Потом он ***начал рассказывать*** мне про игры. [Елена Павлова. Вместе мы эту пропасть одолеем! // «Даша», 2004]

SLINKs are used for subordination connections between event instances. The links are only applicable to event instances with semantic classes REPORTING, I_STATE, AND I_ACTION. E.g.:

(17) Он **отрицает**, что **начал** в Лондоне с выстрела в лицо, — может, потому что Джозеф Зив остался жив. [Сергей Юрьенен. Покер с Ильичом (1997) // «Столица», 01.04.1997]

As mentioned in 4.1, the annotation of semantic classes of events is out of the scope of this paper, and so is the annotation of SLINKs.

TLINKs are meant for establishing relations between temporal elements in the text. The tag can either express a relation between an event and temporal expression or two temporal expressions. For example:

(18) Рукопись итогового труда профессора, над которой он **работал двадцать лет**? [Ю. О. Домбровский. Обезьяна приходит за своим черепом. Пролог (1943-1958)]

The relation between the event instance *работал* and the temporal expression *двадцать лет* in (18) would be annotated via a TLINK tag.

Finally, event instances and temporal expressions are linked via signals. For example, the signal ‘в’ in (19) would be used to link the event instance *появился* and the temporal indicator *час*.

(19) В пансионе Адольф Задер появился **в** час, когда ударил гонг к обеду. [А. Н. Толстой. Черная пятница (1924)]

Signals are a rather small set of prepositions (*до, к, после, в* etc.), and are thus stored in a predefined list. The algorithm iterates over TIMEX3 tags in the text and checks whether the temporal indicator is preceded by a signal. It then uses the results of dependency parsing to find any event instances that are syntactically related to the signal in a way that would allow us to declare a relationship between the event instance and the temporal expression.

4.4. Final Pipeline

In conclusion, the pipeline for annotation consists of the following steps:

1. Named Entity Recognition using Stanza, SpaCy, and Natasha;
2. Dependency Parsing using Stanza, SpaCy, and Natasha;
3. Morphological tagging using Stanza, SpaCy, and Natasha;
4. Extraction of possible events using syntactic and morphological information;
5. Extraction of temporal expressions using the HeidelTime system;
6. Extraction of signals using a predefined set of prepositions, temporal; annotation, and results of dependency parsing;
7. Establishing links and relations between elements of annotation using the results of dependency parsing.

In order to determine the optimal pipeline for the annotation, results for each of the tested models are evaluated on the manually annotated gold standard.

Chapter V. Results

5.1. Evaluation and the optimal pipeline

The pipeline consists of many stages, which in turn can be evaluated in a variety of ways.

In the first step of the pipeline – Event Extraction – the annotation is a result of morphological and syntactic parsing. Therefore, the number of events extracted, their IDs, and attributes vary depending on the model used for extracting morphological features and syntactic information.

First, the number of events in the gold standard and automatically produced annotation are compared. Since the algorithm can produce errors by both extracting event triggers that are not in the gold standard and missing event triggers that are in the gold standard, the evaluation for each model consists of the number of false positives (an event that does not appear in the gold standard was extracted) and false negatives (an event that appears in the gold standard was not extracted). Table 6 shows the number of false positives and false negatives for each model used. The total number of event instances in the gold standard is 355.

	Number of false positives	Number of false negatives
Stanza	17 (4.8%)	70 (19.7%)
SpaCy	**12 (3.4%)**	**68 (19.1%)**
Natasha	30 (8.4%)	70 (19.7%)

Table 6. Number of false positives and false negatives for Stanza, SpaCy, and Natasha.

Since the methodology was mostly aimed at extracting as few ‘false’ events as possible, the number of false negatives is much higher. SpaCy performs better than other models, however, the number of false negatives is quite large even for the best performing model, which suggests that the focus of future algorithm development needs to be on finding other ways to extract less prototypical events, both in terms of syntactic structure and morphology (event triggers that are expressed by parts of speech other than verbs are not extracted).

Although Natasha was trained on Russian data and mostly news texts, the model shows the worst results at this stage of evaluation. Stanza performs almost as well as SpaCy, however, the memory and time efficiency is a serious trade-off for this model. It is thus not reasonable to use Stanza for the final pipeline.

Another stage of the pipeline that utilizes syntactic information provided by the three models is extracting signals. The correct extraction of signals is incredibly important, as they are used to link event instances to temporal expressions. In the case of extracting the signals, false positives are not possible, so the evaluation consists of determining the total number of signals the models have missed. Table 7 shows the number of signals each of the models failed to extract out of 68 signals in the gold standard.

	Stanza	SpaCy	Natasha
Number of signals missed by the model	9 (13%)	**8 (11.7%)**	11 (16%)

Table 7. The number of signals the model failed to extract for Stanza, SpaCy, and Natasha.

Once again, SpaCy shows the best results, although the difference is not significant for such a small number.

5.2. Shortest path between entities

The Shortest Path between Entities (SPE) approach is aimed at extracting events that would have been otherwise missed (less prototypical event instances, for example, those expressed by a noun). However, it is also important to make sure that SPE does not yield too many false positives, i.e. lead to cluttered data where many event instances extracted do not possess any semantic characteristics of events. The main problem with the approach is there is no inherent threshold for when the shortest path is relevant, instead, all syntactic connections between named entities are extracted. The way (Rusu et al. 2014) filters out syntactic connections in SPE is by creating a set of patterns that define most contexts where SPE is applicable. However, defining such patterns is a fundamental work and is not possible in the context of this paper. Additionally, the peculiarities of Russian syntax seem to have a negative effect on the efficiency of such an algorithm. After looking at the results of the SPE approach on our data, it was decided that the algorithm is not suitable for EE in Russian without substantial work in developing patterns and rules for the syntactic connections between entities that can be classified as event instances.

5.3. Final dataset

Thus the pipeline used to annotate the final dataset is the following:

1. Extract syntactic information using the SpaCy dependency parser;
2. Extract morphological information using the SpaCy morphology tagger;
3. Extract temporal expressions using the HeidelTime system;
4. Extract signals using a predefined set of prepositions and the results of dependency parsing;
5. Link temporal expressions to event using signals;
6. Link temporal expressions using syntactic information.

The final dataset consists of 82 individual files with texts annotated by the pipeline described above, where all morphological and syntactic information was extracted using SpaCy that showed the best result in evaluation. The total number of words in the final dataset is 15703, the total number of event instances extracted is 982, the total number of temporal expressions is 443, the total number of signals is 204.

595 (60%) verbs have been annotated as perfective, 387 (40%) as imperfective. 704 (72%) event instances are in the past tense, while 278 (28%) are in the present tense.

Chapter VI. Challenges and future prospects

The process of creating a TimeBank is a complex series of tasks and problems that need to be solved. Each of the stages of the pipeline poses its own challenges, some of which are rooted in the specifics of the TimeML standard, while others are rooted in Russian's complex morphology and syntax.

Firstly, the definitions for the fundamental concepts are not always clear and concise. For example, the definition of an event and the semantic classes of events are very vague, leaving a lot of room for interpretation. Additionally, since the annotation standard was developed for English, other languages may require a different set of values for the 'class' attribute.

Secondly, the motivation behind certain parts of the annotation standard lies in the possible application for the resources that utilize it. For example, generics are not tagged as event instances, although they could be useful for question answering NLP-systems (Saurí et al. 2006). Depending on the applications of the resource, some tags can be omitted or supplemented by including more grammatical or semantic information.

The HeidelTime system used for extraction and normalization of temporal information, though most often accurate, is not perfect as the set of rules for Russian is in the preliminary stage. For example, for the sentence 'We went out for **new year's eve**' in a text with DCT 01/01/2003 it annotates the value of the temporal expression as 12/31/2003, instead of 12/31/2002. A way to correct errors in such contexts is using machine-learning models. Since the value of the temporal expression here depends heavily on the entire context and certain implications (e.g. a person talking about a holiday is more likely to refer to the date that just passed, not next year, unless explicitly stated). Perhaps a combination of the rule-based approach in HeidelTime and a machine-learning model for post-processing and error correction would yield even more accurate results. Additionally, the accuracy of temporal annotation can be improved by utilizing NER models, since HeidelTime detects temporal expressions in Named Entities, e.g.:

(20) <TIMEX3 tid="t5" type="DURATION" value="PT1H">*Час*</TIMEX3> *Суда*.

An issue that has been mentioned in 4.1. is extracting event instances of the same event. There are multiple ways two or more event instances can be expressed by the same event indicator. Firstly, anaphora and coreference, which have been described in 4.1. Secondly, the number of event instances can be conveyed through quantifiers:

(21) Несмотря на **дважды принимавшийся** дождь, люди не расходились, а с интересом наблюдали за развитием действия на экране, а в конце несколько минут аплодировали авторам этого удивительного вечернего шоу. [Александр Смотров. Тысячи людей собрались на Трафальгарской площади , чтобы увидеть новую версию знаменитого фильма «Броненосец Потемкин» (2004) // РИА «Новости», 13.09.2004]

Finally, the most challenging case is elliptical sentences:

(22) Наряду с другими госбанками Сбербанк ***решает*** *не только корпоративные задачи, но и государственные.*

In the example above the two event instances of the indicator *решает* are reflected on the nominal clauses. In other words, in order to extract the event instances, one must also correctly parse the arguments of the verb. Adding a preprocessing step for extracting the verb arguments might facilitate event instance extraction in such cases.

In conclusion, although there are a number of issues that are yet to be solved, the preliminary results that this paper introduces are applicable to most prototypical cases. In addition, the pipeline introduced in Section 5.3. can be used as the basis for TimeBank annotation and completed by manual annotation of complex cases.

Conclusion

The category of time is an important cognitive concept that can be conveyed through a variety of linguistic instruments. Time is relative, and temporal elements tend to be anchored in one another. It is thus challenging to reconstruct all temporal relations in a text, specifically with automatic parsing and annotation.

Parsing temporal information in a text is not a trivial task: temporal indicators (such as ‘tuesday’ or ‘month’) are modified by quantifiers (such as ‘twice’) and modifiers (such as ‘every’). In addition, temporal expressions are introduced by signals (such as ‘on’ or ‘before’), which convey relations between events and temporal expressions.

Temporal information is closely intertwined with occurrences or processes and is most often used to describe an interval or a set of intervals when an event takes place.

TimeML is a universal annotation standard for temporal annotation, event annotation, establishing links and relations. The annotation standard facilitates the creation of temporally annotated corpora which can have a variety of applications in both theoretical and applied linguistics.

In this paper I have introduced an attempt at creating an efficient pipeline for automatic annotation according to the universal TimeML standard.

First, the gold standard was manually annotated as a source for evaluation. Second, event extraction was carried out using a combination of syntactic and morphological analysis. Temporal information was then extracted from the texts using the rule-based HeidelTime system. Finally, information about events and temporal expressions is linked using the results of dependency parsing.

The resulting annotations of each model used for syntactic and morphological analysis were evaluated using the gold standard. The most efficient model was chosen for the final pipeline and used to annotate a greater number of texts.

In conclusion, this paper proposes a pipeline for automatic annotation of the Russian TimeBank, gives an overview of different approaches to extracting and normalizing temporal information, event extraction, and establishing relations between events and temporal expressions. As a result, a temporally annotated corpus of Russian news texts was created, annotated in accordance with the universal TimeML standard.

Appendix 1.

Code and data: [URL: https://github.com/zojabutenko/Russian_TimeBank]

References

Allen J. F. An Interval-Based Representation of Temporal Knowledge //IJCAI. – 1981. – T. 81. – C. 221-226.

Bittar A. et al. French TimeBank: an ISO-TimeML annotated reference corpus //Proceedings of the 49th annual meeting of the Association for Computational Linguistics: Human Language Technologies. – 2011. – C. 130-134.

Caselli T. et al. Annotating events, temporal expressions and relations in Italian: the It-TimeML experience for the Ita-TimeBank //Proceedings of the 5th linguistic annotation workshop. – 2011. – C. 143-151.

Galambos A. Primary and secondary imperfectives in Russian: A cumulativity analysis //LSOWorking Papers in Linguistics 7: Proceedings of WIGL 2007. – 2007. – C. 79-94.

Gast V. et al. Enriching timebank: Towards a more precise annotation of temporal relations in a text //Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16). – 2016. – C. 3844-3850.

Ju T. S. et al. RU-EVAL-2014: Evaluating anaphora and coreference resolution for Russian //Komp’juternaja Lingvistika i Intellektual’nye Tehnologii. – 2014. – C. 681-694.

Klein, W. Time in Language. Routledge, London. - 1994

Klein W. A time-relational analysis of Russian aspect //Language. – 1995. – C. 669-695.

Lakoff, G. (1980). Johnson. M. Metaphors We Live By, U. *Chicago: The University of Chicago Press, 1*(980), 32.

Lyashevskaya O. N. et al. GRAMEVAL 2020 shared task: Russian full morphology and universal dependencies parsing //Proc. of the International Conference Dialogue. – 2020. – T. 2020. – C. 553-569.

Nesset T. Case assignment and image schemas: Russian temporal adverbials //Studies in Language. International Journal sponsored by the Foundation “Foundations of Language”. – 2004. – Т. 28. – №. 2. – С. 285-319.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In Association for Computational Linguistics (ACL) System Demonstrations. 2020.

Pustejovsky J. et al. The timebank corpus //Corpus linguistics. – 2003. – Т. 2003. – С. 40.

Pustejovsky J. et al. Timebank 1.2 documentation //Event London, no. April. – 2006. – С. 6-11.

Rusu D., Hodson J., Kimball A. Unsupervised techniques for extracting and clustering complex events in news //Proceedings of the Second Workshop on EVENTS: Definition, Detection, Coreference, and Representation. – 2014. – С. 26-34.

Saurí R. et al. TimeML annotation guidelines //Version. – 2006. – Т. 1. – №. 1. – С. 31.

Saurı R., Badia T. Spanish TimeBank 1.0 Corpus documentation. – 2012.

Saurı R., Badia T. Catalan timebank 1.0 corpus documentation. – Technical report, 2012.

Schmid H. Probabilistic part-of-speech tagging using decision trees //New methods in language processing. – 2013. – С. 154.

Shavrina T., Shapovalova O. (2017) TO THE METHODOLOGY OF CORPUS CONSTRUCTION FOR MACHINE LEARNING: «TAIGA» SYNTAX TREE CORPUS AND PARSER. in proc. of “CORPORA2017”, international conference , Saint-Petersbourg, 2017.

Starostin A. S. et al. FactRuEval 2016: evaluation of named entity recognition and fact extraction systems for Russian. – 2016.

Strötgen J. et al. KRAUTS: a German temporally annotated news corpus //Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). – 2018.

Strötgen J., Gertz M. Heideltime: High quality rule-based extraction and normalization of temporal expressions //Proceedings of the 5th international workshop on semantic evaluation. – 2010. – C. 321-324.

UzZaman N., Allen J. TRIPS and TRIOS system for TempEval-2: Extracting temporal information from text //Proceedings of the 5th International Workshop on Semantic Evaluation. – 2010. – C. 276-283.

Zell J., Strötgen J. HeidelTime standalone manual version 2.1 //Proceedings of the 5th International Workshop on Semantic Evaluation. – 2015.