Semantic-based alignment algorithm for a Russian-Japanese bilingual parallel corpus

Biryukova E. M.
Linguistic Convergence Laboratory, HSE
Moscow, Russia
kateabr@yandex.ru

Abstract

When compiling a parallel corpus, it is vital to align texts in such a way that semantically parallel sentences are stored together as a cluster. While this task is too time-consuming to be performed manually, already available solutions either cannot be adopted to the Japanese-Russian language pair, or require enormous amounts of already aligned material as a training set. This article presents an aligning algorithm based on semantic pivot comparison, as well as its accuracy rates, and discusses some ways of boosting its accuracy.

Keywords: computer linguistics, corpus linguistics, computer semantics
DOI: 10.28995/2075-7182-2021-20-1041-1049

Семантический алгоритм выравнивания для параллельного русско-японского корпуса[1]

Бирюкова Е. М.
Международная Лаборатория Языковой Конвергенции, НИУ ВШЭ
Москва, Россия
kateabr@yandex.ru

Аннотация

При составлении параллельного корпуса необходимо выровнять тексты относительно друг друга так, чтобы параллельные по своему смыслу предложения хранились вместе, образуя кластер. Эта задача слишком трудоемка для выполнения вручную, а существующие решения либо не могут быть применены для языковой пары «японский-русский», либо требуют для обучения огромного количества уже выровненного материала, практически на данный момент недоступного. В данной статье представлен алгоритм, выполняющий выравнивание при помощи семантических опорных точек, приведена точность его работы и обсуждаются возможные пути его улучшения.

Ключевые слова: компьютерная лингвистика, корпусная лингвистика, компьютерная семантика

1 Введение

Параллельный корпус – это совокупность наборов из двух или более текстов на двух или более языках, где один из текстов обычно является оригиналом, а остальные – его переводами. Подобные корпуса используются в самых разных целях: они могут служить источником информации для обучения алгоритмов при автоматическом анализе естественного языка, они могут быть инструментом для различного рода корпусных исследований, для изучения языков и решения множества других задач.

[1]Исследование поддержано проектом «Компьютерно-лингвистическая платформа нового поколения для цифровой документации русского языка: инфраструктура, ресурсы, научные исследования» (Грант в форме субсидии, Соглашение Минобрнауки РФ от 13 октября 2020 года №075-15-2020-793).

Одним из важнейших этапов формирования таких корпусов является выравнивание текстов на разных языках относительно друг друга. Поскольку ручное выравнивание – очень долгий и трудоемкий процесс, а существующие инструменты выравнивания не могли быть использованы для работы над языковой парой «русский-японский» (подробнее об этом в разделе 2), автором была написана программа, позволяющая выравнивать предложения на данных языках, объединяя предложения в кластеры – семантически параллельные наборы предложений на русском и японском языках; в дальнейшем эта программа была использована в ходе работы над параллельным русско-японским корпусом, предназначенным для включения в НКРЯ [11], а пока доступным в виде демонстрационной версии [12].

2 Обзор предыдущих работ

Для реализации данной программы был выбран лексический метод [2]: вначале тексты дробятся на предложения, а затем слова предложения на одном языке сопоставляются со словами предложения на другом языке[2]. Стратегия сравнения длин предложений и слов в пределах одного предложения при помощи алгоритма Гейла-Черча [4], как это реализовано в HunAlign [5], не могла быть использована в силу графической несопоставимости письменных представлений японского и русского языков[3], хотя, возможно, это впечатление обманчиво, поскольку подобный метод показал неплохую точность на языковой паре «английский-японский» [1] – с некоторыми оговорками. Методы выравнивания, основанные на применении нейронных сетей, такие как [9] и [3], в свою очередь, требуют огромного количества уже размеченных и готовых к использованию в качестве обучающего множества данных и больши́х вычислительных мощностей, поэтому реализация подобного метода на текущий момент не представляется возможной. Еще один метод выравнивания, описанный в [8] и заключающийся в генерации машинного перевода с последующим нахождением пар предложений с оптимальным значением метрики BLEU [7], также не может быть использован в данный момент из-за высоких требований к производительности. Таким образом было решено разработать собственный алгоритм и метрику оценивания близости предложений, принимающие во внимание особенности имеющихся данных.

3 Принцип работы предложенного алгоритма

3.1 Метрика близости предложений

В целях подсчета меры близости для каждого предложения из текущей пары «японский-русский» подсчитываются следующие коэффициенты:

- cnt_{in}: количество опорных точек русского предложения (здесь и далее под опорными точками подразумеваются словоформы одного текста, которым алгоритм пытается сопоставить словоформы другого текста пары), которые удалось при помощи словаря сопоставить с опорными точками японского предложения; когда таких точек нет, полагается равным 0.1 во избежание ошибки, возникающей при попытке деления на ноль;
- cnt_{out}: количество опорных точек, которые не удалось сопоставить с опорными точками предложения на другом языке;
- cnt_{total}: общее количество доступных для сравнения опорных точек без дубликатов (поскольку сопоставление выполняется после приведения слов к начальным формам, для того, чтобы избежать возможного искажения метрики в лучшую сторону за счет повтора одной и той же

[2]Разбиение предложения на слова с последующей лемматизацией и аннотацией полученных единиц выполняется при помощи морфологического анализатора JUMAN++ [15].

[3]К японскому, как и ко всем идеографическим языкам, неприменимо понятие слова как единицы речи с четко выраженными границами [10]; неясно, по какому критерию корректнее подсчитывать длину предложения: по количеству символов в нем самом или его фонетической репрезентации. Более того, если зафиксировать один из этих вариантов и строго придерживаться его при оценке длины, в обоих случаях длины одинаковых предложений, записанных в разном виде, могут оцениваться совершенно по-разному: в первом случае – из-за длины слоговой репрезентации чтения, которая варьируется от иероглифа к иероглифу без какой-либо закономерности; во втором – из-за того, что один и тот же иероглиф может иметь как несколько разных значений, кодируемых чтениями разной длины (напр. « 館 » [yakata]: 'дворец, палаты'; [tachi / tate]: '*то же*, маленькая крепость'), так и несколько разных чтений, кодирующих одно значение (напр. « 茅 » [chi / chigaya]: аланг-аланг [13]).

опорной точки, каждое слово следует учитывать ровно один раз вне зависимости от того, сколько раз оно было употреблено в предложении);

- cnt_{new}: количество опорных точек, полученных из нового предложения и не являющихся дубликатами уже имеющихся.

Среди возможных вариантов сочетания этих параметров оптимальной оказалась следующая формула:

$$S_{noun/verb} = \begin{cases} \frac{penalty_1}{cnt_{in}}, \text{ если } cnt_{total} = 0 \\ \frac{penalty_2}{cnt_{in}}, \text{ если } cnt_{new} = 0 \\ \frac{cnt_{out}}{cnt_{in}}, \text{ если } cnt_{total} \neq 0 \text{ и } cnt_{new} \neq 0 \end{cases} \quad (1)$$

где $penalty_{1,2}$ – коэффициенты, ухудшающие оценку для новых предложений в соответствующих случаях, перечисленных в правой части формулы. Эмпирическим путем были подобраны $penalty_1 = 7$, $penalty_2 = 50$ для японского языка и $penalty_1 = 2$, $penalty_2 = 4$ – для русского. Изменение этих коэффициентов влияет на чувствительность первичного разбиения: чем выше значения этих коэффициентов, тем выше вероятность того, что предложение, принадлежащее кластеру, будет от него ошибочно отделено, и напротив, чем их значения ниже, тем вероятнее предложение, не принадлежащее кластеру, будет к нему прикреплено в случаях, когда в этом предложении не имеется опорных точек или они не являются уникальными в пределах кластера. Столь существенная разница в коэффициентах $penalty_{1,2}$ для русского и японского языков обусловлена стабильно бóльшим набором опорных точек у японских предложений из-за предоставляемых онлайн-словарем ЯРКСИ пояснений и синонимов, которые также нельзя исключать из рассмотрения. Строгое разделение на части речи позволяет несколько снизить уровень возникающего при этом информационного шума: так, для точек-глаголов из словарного перевода извлекаются только глаголы, а для существительных – существительные (на данный момент рассматриваются только эти части речи, поскольку они встречаются в текстах чаще всего). При этом значение метрики для каждой из частей речи подсчитывается и хранится отдельно, а финальное значение является их произведением.

Значение определенной таким образом метрики будет тем меньше, чем больше опорных точек из множества, заданного русским предложением, можно сопоставить с опорными точками японского предложения. Также следует подчеркнуть, что, поскольку метрики для японского и русского предложения подсчитываются независимо, полный вариант расстояния между предложениями включает в себя два значения – для русской и японской половин кластера.

3.2 Выбор алгоритма выравнивания

Целью работы программы является такое разбиение текста на параллельные кластеры, чтобы среднее значение метрики по двум текстам было минимальным, а самих кластеров было как можно больше. У этой задачи два решения, первое −полный перебор, который гарантированно отыщет оптимальное разбиение, но за время, практически экспоненциально возрастающее в зависимости от объема текстов и разницы в их объеме. В самом простом случае, когда количество кластеров равно количеству предложений в более коротком тексте, а максимальное количество предложений более длинного текста, соответствующих одному предложению более короткого текста, полагается равным двум, количество итераций, необходимых этому алгоритму для обработки одной пары текстов, можно подсчитать по формуле сочетаний:

$$C_n^{m-n} = \frac{n!}{(m-n)!(2n-m)!}, \quad (2)$$

где m, n – количество предложений в русском и японском тексте соответственно, удовлетворяющих условию $n < m < 2n$.

Поскольку такая зависимость накладывает ограничения на размер обрабатываемых текстов, было решено реализовать выравнивание на основе описанного ниже алгоритма поиска с отсечением [6], то есть в принципе не рассматривать заведомо проигрышные варианты кластеризации и производные от них. Тогда время работы алгоритма возрастает линейно относительно количества предложений в текстах и разности между бóльшим и меньшим из них, и алгоритм будет возможно использовать для более длинных текстов с большой разницей в количестве предложений. В то же время это значит, что вероятность генерации программой ошибочного разбиения возрастает. Поскольку целью написания данной программы являлось облегчение выравнивания текстов, а не его полная автоматизация, это не так критично.

3.3 Программная реализация алгоритма

В начале своей работы программа выполняет первичное разбиение текстов на кластеры близких между собой предложений. Начинается такое разбиение с подсчета расстояния между парой первых предложений выравниваемых текстов. Затем к этим двум предложениям добавляется еще одно предложение из русскоязычного текста, и расстояние подсчитывается уже для кластера из трех предложений. Если новое значение русского варианта метрики меньше предыдущего, то предложение добавляется к кластеру, а оба значения метрики обновляются соответствующим образом. Таким же образом проверяется принадлежность кластеру следующего японского предложения, затем – еще одного русского и так далее до того, как значение метрики не перестанет убывать. Тогда кластер считается законченным и программа начинает работу над новым кластером.

После того как текст разделен на кластеры, каждый кластер, включающий в себя более чем одно русское или японское предложение, проходит проверку балансировкой, заключающуюся в следующем: если при исключении из текущего кластера первого японского предложения и добавлении его к предыдущему среднее значение метрики для японского языка в этих двух кластерах уменьшается, то подобная конфигурация фиксируется, а значения метрики обновляются уже для обоих языков. Затем аналогичная проверка проводится для последнего японского предложения, затем – для русских, и так далее, пока среднее метрик не перестанет улучшаться. Этот шаг позволяет выровнять предложения более корректным образом в том случае, когда предложения на стыках кластеров короткие или не имеют достаточного количества опорных точек для того, чтобы улучшить показатель близости для своего кластера: тогда оно прикрепляется к следующему кластеру, что негативно сказывается на оценке близости входящих в него предложений.

4 Применение алгоритма на практике

4.1 Корректность работы

Предложенный алгоритм демонстрирует достаточно высокую точность кластеризации: она была корректна в 74% случаев при исключении из рассмотрения случаев, в которых кластера на каком-то из этапов были дестабилизированы настолько сильно, что вернуться к корректному разбиению в последующих кластерах не получилось, и в 65%, если такие случаи из рассмотрения не исключались, причем условием корректности считалась минимальность кластера, то есть наименьшее количество действительно параллельных по своему смыслу предложений; эталоном корректности являлись результаты ручного выравнивания, примененного для построения пилотной версии корпуса (на данный момент включает в себя 47 размеченных параллельных текстов, состоящих из 42984 и 31752 токенов для русского и японского языка соответственно; тексты хранятся в формате .xml-файлов, разметка выполнена автором). В общем же корректность работы алгоритма составляет 81%: именно столько полученных в результате работы кластеров были параллельными, хотя при этом они не всегда являлись минимальными.

4.2 Обсуждение: недостатки предложенного алгоритма

Очевидным недостатком предложенного подхода является его плохая работа на предложениях, в которых нет опорных точек. Если таких предложений несколько, то алгоритм начинает накапливать ошибки, что в конечном итоге приводит к совершенно неправильному разбиению. Такой же

эффект возникает, когда опорные точки в русском предложении по какой-то причине не пересекаются с опорными точками в японском; если таких предложений в одном тексте стоит подряд больше, чем в другом, кластерам не удается стабилизироваться. С этим недостатком можно справиться, если еще на этапе деления на предложения группировать предложения без опорных точек в одно длинное предложение, которое не будет прикреплено ни к одному из соседних кластеров, и тем самым не испортит их значения близости и не дестабилизирует их.

Также в данной реализации материалом для опорных точек в японских текстах служит словарный перевод, который может быть загрязнен огромным количеством синонимов, посторонних слов и повторений, влияющих на оценку опорных слов. Проблема повторений уже решена, а проблему синонимов и посторонних слов можно решить группировкой соответствующих одному слову переводов: тогда опорная точка будет считаться общей, если найден русский эквивалент хотя бы одному из них.

Существенным улучшением в плане производительности также будет отказ от использования онлайн-словаря в пользу его локальной версии; на данный момент `Python`-пакет, включающий в себя базы словарей ЯРКСИ и WARODAI [14], уже разработан и на данный момент проходит тестирование.

Но самым важным недостатком алгоритма является предположение о том, что если слово является той или иной частью речи в исходном тексте, оно будет той же частью речи и в переводе. К примеру, слово « 連絡 » [renraku] ’связь, контакт’ в отрывке, приведенном в следующем разделе, являющееся существительным с морфологической точки зрения, на русский переведено как «связаться», так как это существительное образует глагол добавлением вспомогательного глагола «suru», который в данном случае в тексте по какой-то причине опущен. Среди приведенных в словаре вариантов перевода образуемого им глагола («связываться, соединяться») нужный вариант отсутствует, поэтому отследить взаимосвязь этих двух слов на данном этапе невозможно. Ситуации такого рода, а также более вольное трактование переводчиками исходных текстов, в большей степени свойственны художественным произведениям. Они сильнее всего искажают оценку близости, но к подобным лексическим изменениям устойчив алгоритм Гейла-Черча: возможно, если при его помощи выполнять первичную кластеризацию текстов с дальнейшей балансировкой по придложенному методу, то и последствия подобных проблем удастся минимизировать.

4.3 Пример работы алгоритма

В этом разделе представлен пример работы программы на отрывке из текста, подходящем для иллюстрации обоих его шагов. Для каждого шага также приведена информация, генерируемая на этапе подсчета метрик: количество пересекающихся и непересекающихся опорных точек, значения опорных точек для предыдущего и текущего шагов обработки кластера и значения метрик для русской и японской половин кластера.

(1) По словам Ютани, который родился и вырос в Осаке, примерно в 1960 году они с отцом Кэйдо заехали в старый дом. Отец показал ему кровати и сказал, что на них спали русские, взятые в плен во время русско-японской войны. Из нашей и других газет Ютани узнал, что Кусано исследует историю тех лет на основе фотографий русских военнопленных, снятых в эпоху Мэйдзи. Ютани связался с Кусано. 3 февраля они посетили склад в городе Хакусан и осмотрели кровати и другие предметы.

大阪で生まれ育った油谷さんは 1960（昭和 35）年ごろ、父の奎道さんと旧松任町の実家に帰省した際、蔵にある古いベッドを見せられ、「日露戦争の捕虜が使ったものだ」と説明を受けたという。油谷さんは、本紙報道などで、草野さんが明治期に撮影されたロシア人捕虜の写真を頼りに当時の足跡などを調べていることを知り、草野さんに連絡。2 人で 3 日、白山市の蔵を訪れ、片づけてあったベッドなどを調べた。

Корректный вариант выравнивания текстов:

(2) *Кластер 1:*
По словам Ютани, который родился и вырос в Осаке, примерно в 1960 году они с отцом

Кэйдо заехали в старый дом. Отец показал ему кровати и сказал, что на них спали русские, взятые в плен во время русско-японской войны.

大阪で生まれ育った油谷さんは 1960（昭和 35）年ごろ、父の奎道さんと旧松任町の実家に帰省した際、蔵にある古いベッドを見せられ、「日露戦争の捕虜が使ったものだ」と説明を受けたという。

(3) *Кластер 2:*
Из нашей и других газет Ютани узнал, что Кусано исследует историю тех лет на основе фотографий русских военнопленных, снятых в эпоху Мэйдзи. Ютани связался с Кусано.

油谷さんは、本紙報道などで、草野さんが明治期に撮影されたロシア人捕虜の写真を頼りに当時の足跡などを調べていることを知り、草野さんに連絡。

(4) *Кластер 3:*
3 февраля они посетили склад в городе Хакусан и осмотрели кровати и другие предметы.

2 人で 3 日、白山市の蔵を訪れ、片づけてあったベッドなどを調べた。

Вариант разбиения, генерируемый программой на первом шаге работы:

(5) *Кластер 1:*
По словам Ютани, который родился и вырос в Осаке, примерно в 1960 году они с отцом Кэйдо заехали в старый дом.

大阪で生まれ育った油谷さんは 1960（昭和 35）年ごろ、父の奎道さんと旧松任町の実家に帰省した際、蔵にある古いベッドを見せられ、「日露戦争の捕虜が使ったものだ」と説明を受けたという。

```
JA:VERB 0/49 | [] + []
JA:NOUN 0/49 | [] + ['год', 'дом', 'отец', 'осака']
RU:VERB 0/3  | [] + []
RU:NOUN 4/3  | [] + ['дом', 'отец', 'осака', 'год']

                RU:21.951     JA:3226.829
```

Нотацию при этом следует понимать следующим образом:

- JA, RU – язык опорных точек;
- NOUN, VERB – часть речи, к которой принадлежат опорные точки;
- 0/49 – отношение количества совпавших опорных точек к количеству не совпавших;
- [] + ['год', 'дом', 'отец', 'осака'] – множество уже имеющихся в кластере общих опорных точек, к которому добавляется множество новых опорных точек, построенное на текущем шаге.

Значения коэффициентов при этом составили:

- для японского текста:
  - cnt_{in}: 0.1, cnt_{out}: 49, cnt_{total}: 49, cnt_{new}: 49 – для глаголов;
  - cnt_{in}: 4.1, cnt_{out}: 27, cnt_{total}: 31, cnt_{new}: 31 – для существительных;
- для русского текста:
  - cnt_{in}: 0.1, cnt_{out}: 3, cnt_{total}: 3, cnt_{new}: 3 – для глаголов;
  - cnt_{in}: 4.1, cnt_{out}: 3, cnt_{total}: 7, cnt_{new}: 7 – для существительных.

Таким образом, поскольку ни cnt_{total}, ни cnt_{new} не равны нулю, итоговое значение метрики для данного кластера составляет:

$$\begin{cases} S_{ja} = S^{verb} * S^{noun} = \dfrac{cnt_{out}^{verb}}{cnt_{in}^{verb}} * \dfrac{cnt_{out}^{noun}}{cnt_{in}^{noun}} = \dfrac{49}{0.1} * \dfrac{27}{4.1} = 3226.829268292683 \\ S_{ru} = S^{verb} * S^{noun} = \dfrac{cnt_{out}^{verb}}{cnt_{in}^{verb}} * \dfrac{cnt_{out}^{noun}}{cnt_{in}^{noun}} = \dfrac{3}{0.1} * \dfrac{3}{4.1} = 21.951219512195124 \end{cases} \tag{3}$$

Теперь добавим к кластеру еще одно русское предложение:

(6) *Кластер 1:*
По словам Ютани, который родился и вырос в Осаке, примерно в 1960 году они с отцом Кэйдо заехали в старый дом. Отец показал ему кровати и сказал, что на них спали русские, взятые в плен во время русско-японской войны.
大阪で生まれ育った油谷さんは 1960（昭和 35）年ごろ、父の奎道さんと旧松任町の実家に帰省した際、蔵にある古いベッドを見せられ、「日露戦争の捕虜が使ったものだ」と説明を受けたという。

```
JA:VERB 2/47 | [] + ['показывать', 'сказать']
JA:NOUN 8/24 | [] + ['дом', 'год', 'отец', 'война',
                     'плен', 'осака', 'время']
RU:VERB 2/4  | [] + ['сказать', 'показывать']
RU:NOUN 7/4  | ['дом', 'отец', 'осака', 'год'] +
               + ['время', 'плен', 'война']

              RU:21.951 → RU:1.073 ▼
```

Этот вариант более удачен, поэтому предложение закрепляется за кластером.

(7) *Кластер 2:*
Из нашей и других газет Ютани узнал, что Кусано исследует историю тех лет на основе фотографий русских военнопленных, снятых в эпоху Мэйдзи.
油谷さんは、本紙報道などで、草野さんが明治期に撮影されたロシア人捕虜の写真を頼りに当時の足跡などを調べていることを知り、草野さんに連絡。

```
JA:VERB 2/27 | [] + ['узнавать', 'исследовать']
JA:NOUN 4/33 | [] + ['газета', 'эпоха',
                     'фотография', 'мэйдзи']
RU:VERB 2/0  | [] + ['исследовать', 'узнавать']
RU:NOUN 4/4  | [] + ['мэйдзи', 'фотография',
                     'газета', 'эпоха']

              RU:0.0     JA:103.484
```

Для этого кластера метрика по русскому предложению равна нулю, поскольку присутствуют все необходимые глаголы. Из-за этого она немного ухудшается при добавлении к нему следующего предложения:

(8) *Кластер 2:*
Из нашей и других газет Ютани узнал, что Кусано исследует историю тех лет на основе фотографий русских военнопленных, снятых в эпоху Мэйдзи. Ютани связался с Кусано.
油谷さんは、本紙報道などで、草野さんが明治期に撮影されたロシア人捕虜の写真を頼りに当時の足跡などを調べていることを知り、草野さんに連絡。

```
RU:VERB 2/1  | ['узнавать', 'исследовать'] + []
RU:NOUN 4/4  | ['газета', 'мэйдзи',        + []
                'эпоха', 'фотография']

              RU:0.0     RU:0.464 ▲
```

Таким образом, это предложение становится началом нового кластера:

(9) *Кластер 3:*
Ютани связался с Кусано.
2 人で 3 日、白山市の蔵を訪れ、片づけてあったベッドなどを調べた。

```
JA:VERB 0/21 | [] + []
JA:NOUN 0/4  | [] + []
RU:VERB 0/1  | [] + []
RU:NOUN 0/1  | [] + []

             RU:100.0     JA:8400.0
```

Добавление следующего предложения улучшает оценку кластера, поэтому оно считается принадлежащим ему:

(10) *Кластер 3:*
Ютани связался с Кусано. 3 февраля они посетили склад в городе Хакусан и осмотрели кровати и другие предметы.
2 人で 3 日、白山市の蔵を訪れ、片づけてあったベッドなどを調べた。

```
JA:VERB 1/20 | [] + ['посещать']
JA:NOUN 1/3  | [] + ['склад']
RU:VERB 1/2  | [] + ['посещать']
RU:NOUN 1/5  | [] + ['склад']

             RU:100.0 → RU:8.264 ▼
```

На этапе балансировки первый кластер считается удачным:

$$mean(1.073, 38.71) = 19.892 > mean(0.116, 145.161) = 72.639 \quad (4)$$

поэтому он остается без изменений. А первое предложение третьего кластера переносится во второй, поскольку оно сильно портит оценку своему текущему кластеру:

$$mean(3.306, 0.465) = 1.886 < mean(8.264, 0.0) = 4.132 \quad (5)$$

Таким образом, в итоге отрывок получается корректно выровненным и полностью совпадающим с приведенным ранее образцом.

5 Заключение

В настоящей статье предложен алгоритм, позволяющий с высокой точностью объединять предложения на русском и японском языке в параллельные кластеры. После доработки и исправления проблем, обозначенных в разделе 4.2 (или минимизации их негативного влияния на результат), алгоритм может быть использован для подготовки текстов к включению их в японско-русский раздел параллельного корпуса НКРЯ.

References

[1] Aligning Parallel Texts : Do Methods Developed for English-French Generalize to Asian Languages? / Kenneth Church, Ido Dagan, William Gale et al. — 1993. — 01.

[2] Aswani Niraj, Gaizauskas Rob. A hybrid approach to align sentences and words in English-Hindi parallel corpora. — 2005. — 06. — P. 57–64.

[3] Bansal Mohit, DeNero John, Lin Dekang. Unsupervised Translation Sense Clustering // Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — Montréal, Canada : Association for Computational Linguistics, 2012. — Jun. — P. 773–782. — Access mode: https://www.aclweb.org/anthology/N12-1095.

[4] Gale William A., Church Kenneth W. A Program for Aligning Sentences in Bilingual Corpora // Computational Linguistics. — 1993. — Vol. 19, no. 1. — P. 75–102.

[5] HunAlign. — Access mode: https://opus.nlpl.eu/letsmt-dev/doc/LetsMT/Align/Hunalign.html.

[6] Knuth Donald E., Moore Ronald W. An analysis of alpha-beta pruning // Artificial Intelligence. — 1975. — Vol. 6, no. 4. — P. 293 – 326.

[7] PAPINENI K. BLEU : a method for automatic evaluation of machine translation // 40th Annual meeting of the Association for Computational Linguistics, 2002. — 2002. — P. 311–318.

[8] Sennrich Rico, Volk Martin. Iterative, MT-based Sentence Alignment of Parallel Texts. — 2011. — 05.

[9] Yasuda Keiji, Sumita Eiichiro. Method for Building Sentence-Aligned Corpus from Wikipedia. — 2008. — 01.

[10] В. М. Алпатов П. М. Аркадьев В. И. Подлесская. Теоретическая грамматика японского языка: [В 2-х кн.]. — Наталис, 2008. — Vol. 1.

[11] Национальный корпус русского языка. — Access mode: https://ruscorpora.ru/.

[12] Пилотная версия параллельного русско-японского корпуса «ПРЯНИК». — Режим доступа: https://corpus-demo.herokuapp.com/search.

[13] Смоленский В. Японско-русский компьютерный словарь иероглифов ЯРКСИ (7.7.1). — Access mode: https://www.susi.ru/yarxi/.

[14] Японско-русский электронный словарь Warodai. — Режим доступа: https://warodai.ru.

[15] 日本語形態素解析システム JUMAN++. — Access mode: http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++.