**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования**

**Национальный исследовательский университет**
**"Высшая школа экономики"**

**Факультет гуманитарных наук**

**Образовательная программа**
**«Компьютерная лингвистика»**

**КУРСОВАЯ РАБОТА**

На тему:

# Особенности автоматической сегментации языков с отсутствием границ между словами на письме (на примере тайского языка)

*Название темы на английском:*

# Processing Languages without Word Boundaries: Segmentation in Thai Language

Студентка 1 курса
магистратуры
группы №62/л
Мотина Надежда Евгеньевна

Научный руководитель:
Толдова Светлана Юрьевна
доцент

Москва, 2015 г.

# Введение

Для большинства приложений по обработке естественного языка на начальном этапе необходимо разделить текст на токены - последовательности символов. Это необходимо для последующего морфологического, синтаксического и семантического анализа. Поэтому один из основных начальных этапов обработки текста - токенизация или разделение текста на отдельные слова. Это довольно непростая задача. В языках, основанных на латинском алфавите, например, в английском, слова принято разделять пробелами и знаками препинания. Обычно пробел является достаточным основанием для разделения слов, однако, не всегда. Есть языки без пробелов, но с другими маркерами границ слов. Например, в арабском языке на письме выделяется инициаль и финаль (первая и последняя буква слова), что позволяет однозначно определить начало и конец слова. Во вьетнамском языке выделяются границы слогов, но остается проблема выделения многосложных слов.

Особенную сложность для автоматической обработки представляют языки без разделителей между словами и без каких-либо маркеров границ, причем слова не совпадают со слогами (в языках, где есть многосложные слова). К таковым относятся многие азиатские языки, например, китайский, японский, корейский, тайский, лаосский, кхмерский и другие менее распространенные. В этих языках есть разделители между предложениями или их смысловыми сегментами-фразами, однако нет маркеров границ слов. Из-за отсутствия в тексте пробелов или других признаков, явно указывающих

на границы слов, появляется неоднозначность в разбиении текста на токены. При этом в китайском языке каждый иероглиф означает слог, который можно рассматривать как морфему, что несколько упрощает задачу, а тайская письменность состоит из последовательности согласных и гласных букв из алфавита, и границы слогов определяются правилами.

Для токенизации таких языков было разработано множество методов, которое можно разделить на две группы: методы, основанные на словарях, и методы, основанные на машинном обучении.

В данной работе рассматриваются особенности автоматической обработки языков с отсутствующими границами между словами на примере тайского языка.

## Актуальность темы

Один из самых главных этапов обработки текста на естественном языке - его сегментация на смысловые части: предложения и слова. Однако, когда в языке отсутствуют разделители, это становится нетривиальной задачей. Письменность тайского языка такова, что пробелы между словами не ставятся (только на границах смысловых высказываний или в местах пауз речи). Кроме того, отсутствие пробелов - не единственная проблема, которая усложняет сегментацию. В связи с этим, интересно изучить все особенности тайского языка, найти признаки, по которым можно было бы определить границы слов, и разработать методы, позволяющие успешно использовать эти признаки и правильно сегментировать текст на языке без маркеров границ.

## Цель работы

Изучить особенности сегментации тайского текста на слова и предложения, исследовать существующие методы и инструменты и провести сравнительный анализ. Определить, какие признаки (атрибуты) могут помочь определять границу слова, предложить способы улучшения методов токенизации для тайского языка.

## Задачи

Существует несколько систем обработки текстов на тайском языке, однако, пока не проводилось сравнительного анализа данных систем и их методов. Поэтому можно выделить следующие задачи:

- изучить существующие методы и инструменты сегментации текстов на тайском языке
- найти и исследовать корпусы тайского языка
- протестировать готовые методы и инструменты на едином множестве текстов, оценить их качество, провести сравнительный анализ
- проанализировать ошибки систем токенизации
- выделить множество признаков, релевантных для разделения на слова и используемых в разных системах
- предложить пути улучшения работы методов

# Глава 1. Тайский язык

## 1.1. Особенности письменности и грамматики тайского языка

Тайский язык пока не так хорошо исследован по сравнению с европейскими языками или более распространенными азиатскими. Он относится к группе тайско-кадайской языковой семьи и имеет аналитический грамматический строй. Это значит, что в тайском языке нет словоизменения, а грамматические отношения между словами выражаются с помощью синтаксических средств — учитывается порядок слов, наличие специальных служебных слов и контекст.

В тайском языке используется своя письменность, которая берет начало от древних языков пали и санскрита.

В основном алфавите 44 согласных буквы (при этом 2 из них уже не используются на письме) и 2 согласные вне алфавита (ฺ, ฦๅ). Все согласные делятся на 3 класса: высокий, средний и низкий. От класса первой буквы в слоге зависит то, каким тоном нужно читать слог. В начале слога могут встречаться сочетания согласных (кластеры), но таких сочетаний ограниченное количество - всего 18. Кроме того, есть 4 символа, обозначающих сочетание “гласный+согласный” (ไ ใ ำ). Некоторые согласные не могут стоять в конце слога.

Гласные не входят в алфавит. Всего есть 12 знаков для обозначения гласных звуков (монофтонгов и дифтонгов). На письме гласная буква может располагаться слева, справа, сверху и снизу относительно согласного, после которого она произносится.

Также есть 4 диакритических знака, обозначающих тон, и 3 особых символа (ๆ - обозначает повтор предыдущего слова, ฯ - сокращение предыдущего слова, ์ - значок “каран”, означающий, что согласная под ним не читается). Нет понятия строчных и прописных букв. Также есть тайские цифры, хотя арабские цифры тоже достаточно широко распространены в Таиланде. Знаки препинания в тайском языке обычно не ставятся, хотя в последнее время они часто встречаются в тайских социальных сетях.

Тайский язык - тоновый, используются пять тонов: обычный, низкий, падающий, высокий и восходящий. Слог читается определенным тоном в зависимости от класса первой буквы в слоге, диакритического знака, долготы гласной, а также открытости или закрытости слога.

Порядок слов в тайском языке прямой: подлежащее + сказуемое + дополнение + обстоятельства (обстоятельства иногда могут ставиться в начало предложения). В вопросительных предложениях вопросительное слово ставится на место члена предложения, к которому задается вопрос. В общем вопросе в конце ставится вопросительная частица.

Весьма спорна классификация тайских слов на части речи. Одно и то же слово иногда может означать различные части речи в зависимости от своего положения в предложении и контекста. Например, нет морфологической разницы между прилагательными и наречиями. Также можно понять лишь из контекста, является ли слово глаголом или предлогом, существительным или классификатором. Подобного мнения придерживаются и таиландские лингвисты. “Не существует твердых и неизменных правил, по которым тайские слова относились бы к той или иной части речи, — пишет Прайя Ануман Ратчатон. — Любое из них может быть существительным,

прилагательным, наречием и т.п. в зависимости от позиции в предложении” (Морев Л.Н., 1964). В итоге в разных работах по тайской лингвистике авторы предлагают различные способы классификации тайских слов. Однако нельзя сказать, что любое слово может использоваться в качестве любой части речи. Есть слова, которым можно однозначно приписать определенную часть речи. Также существуют морфемы, которые указывают на принадлежность той или иной части речи.

Словообразование в тайском языке происходит с помощью слово- и слогосложения, аффиксации и редупликации. Повтор существительного означает множественное число, прилагательного или наречия - усиление и интенсивность, глагола - длительность действия (Осипов Ю.М., 1969).

## 1.2. Неоднозначность сегментации в тайском языке. Что считать границей слова?

Фундаментальной семантической единицей в языке является слово. Само понятие “слово” несколько размыто, поэтому иногда довольно сложно сравнить эффективность различных систем токенизации текста. В разных языках под понятием “слово” можно понимать фонологическое слово, ортографическое слово или лексическую единицу (лексему).

Когда в быстрой устной речи сложно услышать паузы между словами, человек неявно распознает слова, основываясь прежде всего на своих суждениях. В некоторых случаях это ведет к неоднозначности разделения на слова. Для иллюстрации проблемы отсутствия пауз/пробелов в статье (Virach Sornlertlamvanich, Tanapong Potipiti, Chai Wutiwiwatchai and Pradit Mittrapiyanuruk, 2000) используется хороший пример на английском языке без пробелов. В данном примере видно, что разные варианты сегментации строки имеют прямо противоположные значения:

“GODISNOWHERE”

1. God is now here.

2. God is no where.

3. God is nowhere.

В тайском языке пробелы ставятся только на границах предложений и отдельных смысловых частей предложения. Между словами пробелов нет. Именно это и представляет трудность для токенизации.

Еще одна проблема - слова, образованные с помощью словосложения.

Иногда значение нового слова складывается из его частей, но есть составные слова, которые очень слабо связаны со значениями составных частей. То есть два слова по отдельности имеют свои значения, а сложенные в одно слово - приобретают совершенно новый смысл. Например, หายใจ (дышать) = หาย (терять) + ใจ (сердце). При этом все эти слова (как части, так и составные) присутствуют в словаре. Получается неоднозначность и требуется дополнительный контекстный (или семантический) анализ.

Тайский лингвист Чайчароен в своей статье (Chaicharoen, N., 2002) предлагает считать основным критерием для тайского слова его непрерываемость (uninterruptability). Единство формы и значения не позволяет разделить слово, не изменив его смысл. Чайчароен считает, что нельзя делить составное слово, если его части очень тесно связаны и по отдельности имеют совсем другое значение, нежели сложенные вместе.

Например, слово แม่น้ำ (река) нельзя разделить на части แม่ (мать) и น้ำ (вода), так как смысл теряется. А вот слово พ่อแม่ (родители) можно разделить на части พ่อ (отец) и แม่ (мать), так как связь между словами слабая. Однако этот критерий весьма субъективен, потому что всё зависит от того, насколько сильна эта связь. Например, непонятно - надо ли делить слово คนขายของ (продавец) на части คน (человек) + ขาย (продавать) + ของ (вещи). В принципе, связь не сильная и смысл сохраняется, но получается, что 3 слова вместе обозначают одно понятие, которое в других языках выражается с помощью одного слова. Поэтому иногда даже в ручной сегментации и разметке корпуса возникают несоответствия (inter-annotator agreement).

Иногда словосочетание и сложное слово практически неразличимы. Это характерно не только для тайского, но и для некоторых других языков,

например, китайского. В такой ситуации одну и ту же строку в зависимости от контекста можно сегментировать по-разному. Например (пример из статьи Wirote Aroonmanakun, 2007) , возьмем два предложения:

1. คนขับรถนั่งคอยอยู่ในรถ (Водитель сидит ждёт в машине.)
2. คนขับรถผ่านแยกนี้ไม่มากนัก (Люди не часто проезжают на машине этот перекресток.)

В первом случае คนขับรถ (водитель) - это одно неделимое слово, а во втором คน (человек) + ขับ (водить) + รถ (машина) - разные слова, так как речь идет не о конкретном человеке, а о людях за рулём в целом.

Помимо семантического критерия тайский лингвист Арунманакун (Aroonmanakun) предлагает синтаксический критерий. Если в составном слове есть части, к которым относятся другие члены предложения, то такие составные слова следует делить на части. То есть, если часть составного слова может объединяться в словосочетание с другим словом, то её можно отделить, как самостоятельное слово.

Иногда в тайских текстах встречаются сегменты на других языках, в основном, на английском. Это тоже необходимо учитывать при токенизации.

## 1.3. Сегментация тайского текста на предложения

Наравне с токенизацией деление на предложения - основной начальный этап обработки текста. Обычно конец предложения на письме явно указывается с помощью знаков препинания и заглавных букв. Но в тайском языке не используются знаки препинания, а пробел не всегда означает конец предложения, поэтому определить границы довольно сложно. С одной стороны, пробел ставится в конце предложения вместо знака препинания, но пробелы также встречаются и внутри предложений, когда нужно отделить одну мысль от другой. Часто пробел отражает ритмическое строение или интонацию в предложении. Возможно, это пришло из тайского стихосложения, где предложения делились на строфы.

Само понятие “предложение”, как и “слово”, определено неоднозначно. Предложение - это грамматически организованная комбинация слов, обладающая смысловой и интонационной законченностью. Обычно предложение представляет из себя одну или несколько грамматических основ (клауз), которые могут быть независимыми или зависимыми.

Наиболее эффективный подход к разделению предложений в тайском тексте - это использование параллельных корпусов тайского и английского языков. Можно использовать информацию о границах предложений в английском тексте, чтобы определить границы соответствующих им тайских предложений (sentence alignment).

# Глава 2. Обзор методов и ресурсов сегментации тайского языка

## 2.1. Исследования по токенизации тайского языка

Есть множество работ, посвященных сегментации текстов и морфологическому анализу на китайском, японском и тайском языках.

Исследования в области автоматической обработки тайского языка начались еще в 1980-ых гг. Например, в 1986 году Пувараван (Yuen Poowarawan, 1986) предложил использовать алгоритм Longest matching algorithm для словарного метода. В 1993 году Сонлетламванит (Sornlertlamvanich) придумал алгоритм Maximum matching algorithm, где текстовая строка делится на все возможные комбинации слов из словаря, а затем алгоритм выбирает ту комбинацию, где получилось наименьшее количество токенов.

Затем возросла популярность вероятностных методов и их стали применять в том числе для сегментации тайских слов. Впервые эти методы испытал на тайском языке Понпрасеткун (A. Pornprasertkul, 1994), используя скрытые марковские модели. Для оптимизации вычислений он использовал алгоритм Витерби, а в качестве атрибутов - грамматические тэги. Алгоритм сегментации текста без пробелов на токены с помощью уравнения Витерби находит вариант сегментации строки, имеющий наибольшую вероятность.

В 1997 году группа лингвистов (A. Kawtrakul, C. Thumkanon, P. Varasarai and M. Suktarachan, 1997) разработала технику выбора оптимального

разделения слов, основанную на триграммных марковских моделях. В 1997 году Мекнавин с соавторами (S. Meknavin, P. Charoenpornsawat and B. Kijsirikul, 1997) предложили методы сегментации с помощью машинного обучения и алгоритмов Winnow и Ripper, в качестве атрибутов используя статистику частеречных n-грамм и частотности слов.

Позже (Virach Sornlertlamvanich, Tanapong Potipiti and Thatsanee Charoenporn, 2000) был использован алгоритм C4.5, автоматически определяющий границы слов на основе корпуса. Основные признаки для обучения алгоритма - информационная энтропия[1] и контекстная информация. С помощью этого алгоритма была получена точность 80% и полнота 50%.

В 2001 году Тхирамунконг и Усанавасин (Theeramunkong, Usanavasin, 2001) предложили свой метод, который предполагал сначала разделить строку на неделимые слоги, следуя набору правил, затем извлечь из получившихся слогов признаки, “склеить” слоги в слова и построить дерево решений, определяющее, является ли полученная строка словом.

Позднее этот метод усовершенствовал Арунманакун (*Aroonmanakun, W., 2002*). Он предложил “склеивать” слоги в зависимости от их коллокаций[2].

Из более современных работ стоит отметить исследование Крыангкрай и соавторов (Canasai Kruengkrai, Virach Sornlertlamvanich, Hitoshi Isahara, 2006), в котором используется метод условных случайных полей (Conditional Random Field или CRF) для разработки модели. В работе предложено два способа: алгоритм Витерби и метод доверительного интервала. В наборе

[1] мера неопределённости или непредсказуемости информации, неопределённость появления какого-либо символа первичного алфавита (https://ru.wikipedia.org/wiki/Информационная_энтропия)

[2] Коллокации - это сочетания (биграммы), в которых 2 элемента встречаются чаще, чем случайно. Это оценивается некоторыми стандартными метриками, например, критерием Стьюдента (t-score), взаимной информацией и т.п.

признаков для машинного обучения авторы учитывали частеречную разметку (POS-tagging). Поэтому, если разметка корпуса недостаточно точна, это может плохо отразиться на качестве предсказаний модели.

## 2.2. Корпусы тайского языка

Как правило, все лингвистические исследования и разработка методов обработки естественных языков начинаются с формирования достаточно большого корпуса - базы из подобранных и обработанных по определенным правилам текстов. Особенно хорошо, если в корпусе имеется подробная разметка частей речи (POS-tagging), синтаксическая разметка. Это необходимо для статистического анализа и проверки эффективности разрабатываемых методов.

В связи с тем, что тайский язык исследован сравнительно мало, особенно в компьютерной лингвистике, размеченные корпуса сложно найти и размер их небольшой.

### *Корпус ORCHID*

Практически во всех статьях об автоматической обработке текстов на тайском языке использовался корпус ORCHID (Virach Sornlertlamvanich, Thatsanee Charoenporn, Hitoshi Isahara, 1999). Это первый корпус, собранный совместными усилиями тайской организации NECTEC и японской исследовательской лаборатории Communications Research Laboratory. Он довольно небольшого размера - 2 Мб (примерно 400,000 слов).

В корпусе присутствует деление на токены и частеречная разметка (POS-теги). Все POS-теги делятся на 14 категорий и 47 подкатегорий (есть отдельные категории для классификаторов и префиксов)[3]. Токенизация и разметка сделаны с помощью вероятностной модели триграмм на основании меньшего корпуса, размеченного вручную. Вероятность высчитывается по следующей формуле:

$$P(W,T) = \prod_{i=1}^{n} P(t_i|t_{i-1},t_{i-2}) * P(w_i|t_i)$$

где $T$ - последовательность POS-тегов $t_1 \ldots t_n$ и $W$ - последовательность слов $w_1 \ldots w_n$. Наиболее вероятная цепочка тегов высчитывается с помощью алгоритма Витерби, затем вероятности ранжируются.

Чтобы снизить вычислительные затраты на обработку каждого варианта сегментации, сначала текст вручную делился на предложения по единой инструкции и затем делился на слоги с помощью правилового метода. Так как вероятностная модель триграмм не дала 100% точности и полноты, потребовалась ручная корректировка границ слов и POS-тегов.

Для каждого текста в корпусе есть также мета-разметка, в которой указан автор, заголовок и источник текста на тайском и английском языках. Большую часть текстов составляют рабочие документы, собранные за 6 лет работы организации NECTEC. В этом главный недостаток корпуса, так как тексты в нем довольно специфические и относятся к одной тематике. Поэтому сложно оценивать работу алгоритмов, обученных на этом корпусе.

[3] Полный список тегов описан в статье Virach Sornlertlamvanich, Thatsanee Charoenporn, Hitoshi Isahara, ORCHID: Thai Part-Of-Speech Tagged Corpus (таблица 5).

*Корпус InterBEST'2009*

Этот корпус - самый большой из имеющихся - 5,036,228 слов, 509 документов, из них 96 новостных текстов, 108 текстов из энциклопедии, 107 рассказов и 198 статей (Release 1, 1st June 2009). Все тексты разделены по отдельным файлам формата .txt, слова разделены знаком "|", параграфы - "\n", пробелы в тексте сохранены как в оригинале. Кроме того есть специальные маркеры для именованных сущностей, аббревиатур и стихотворений (обычно в статьях такие участки рассматривались как 1 токен). В качестве именованных сущностей рассматриваются только имена людей, названия организаций и географические названия. Используется кодировка UTF-8.

В корпусе нет частеречной разметки - только деление на слова. Основной принцип сегментации в корпусе - деление текста на как можно меньшие токены без потери смысла. То есть, если при делении определенного участка смысл высказывания сохраняется, то этот участок делится на меньшие части. Это отличает корпус InterBEST'2009 от ORCHID, где большинство составных слов не делятся. Например, в InterBEST'2009 такие приставки, как การ, ความ, อย่าง, ก้าน, ข้อ, คน, เครื่อง, คำ, ช่าง, ชาว, นัก, ผ้า и т.д. отделяются в отдельный токен, потому как не меняют значение слова, а добавляют его. Если значение составного слова сильно отличается от значений составляющих его частей, то слово не делится. Различные виды животных и растений, имеющие одно и то же начало, тоже не делятся (ปลาดุก, ปลากระดี่, ปลากัด, ปลาหมึก). Еще один признак, по которому сегментированы слова - их непрерываемость. То есть, если между составными частями слова можно вставить зависимое, то такое слово делится на составляющие.

В тайском языке часто встречаются повторы слов, на письме отмеченные знаком ๆ. В большинстве случаев повтор не сильно меняет смысл (добавляя значение множественности, скорости или интенсивности). В таких случаях знак ๆ отделяется, как самостоятельное слово. Однако есть такие сочетания, в которых отделение символа ๆ меняет смысл. В таких случаях повторяется не всё слово, а лишь его часть, и знак ๆ входит в состав слова, например, ทั่วๆไป (в общем), ทั้งๆที่ (не смотря на), ต่างๆนานา (различные, несколько).

Примечательно то, что пословицы и поговорки в корпусе не делятся на слова, а остаются единой строкой текста. Авторы объясняют это тем, что при сегментации на слова, такие выражения теряют свой смысл.

В тайском языке очень много заимствованных слов, которые также могут быть составными. Составители корпуса по-разному расценивают слова, заимствованные из разных языков. Если слово заимствовано из пали-санскрита, то даже если оно составное, в корпусе это слово не разделяется. А слова, заимствованные из европейских языков (в основном, из английского), делятся так же, как в оригинале (например, ฟุตบอล | ทีม = football | team).

Знаки пунктуации, иногда встречающиеся в текстах (чаще всего это кавычки или скобки) выделяются в отдельные токены. Точки, использующиеся в сокращениях тайских слов, тоже отделяются. Веб-сайты и электронные адреса не делятся.

## 2.3. Обзор методов сегментации тайского языка

Существует три основных группы методов для сегментации текста в тайском языке, основанные на правилах, словарях и машинном обучении.

### *2.3.1. Методы, основанные на правилах*

В 1983 году тайский лингвист Чамиапонпонг (Chamyapornpong S., 1983) сформулировал, по каким правилам должны делиться тайские слова. Сначала текст делится на строки по пробелам, а далее в строках каждому символу приписывается вероятность того, что он является границей слова. Все символы делятся на 5 групп:

1. Неделимые символы, такие как อั, อิ, อุ, อ็ и อ่.
2. Начальные символы, такие как เ, แ, ไ, ใ и โ.
3. Последующие символы (follower), такие как ะ, า и ำ.
4. Значок “каран” อ์, который оглушает согласную, стоящую под ним.
5. Остальные символы.

Правила сегментации составляются в зависимости от группы. Например, символ четвертой группы обычно стоит в конце слова.

Недостаток такого подхода в том, что необходимо вручную написать очень много правил, а точность и полнота метода остаются низкие. Тем не менее, в 2006 году Уратхамакун и Рунапонгса (Payothorn Urathamakun and Kanda Runapongsa, 2006) написали более гибкие, обновленные правила, учитывающие новые слова, заимствованные из английского и других языков.

В связи с глобализацией и развитием международных связей английский язык активно проникает в тайский, и многие английские слова транслитерируются тайскими буквами. Таких слов всё больше, и их активно используют в речи с характерным падающим или высоким тоном в конце слова.

Хотя метод малоэффективен для токенизации, он широко используется для деления тайского текста на слоги, как один из первых этапов других методов сегментации, и показывает хорошую точность.

### *2.3.2. Словарные методы*

Словарные методы используют наборы термов из словаря для парсинга и сегментации входящей строки на токены. Эффективность словарных методов сильно зависит от качества и размера словарей. Эти методы относительно просты и прямолинейны. Однако, есть две проблемы. Во-первых, незнакомые слова, отсутствующие в словаре. Эту проблему можно решить постоянным расширением словаря и добавлением новых слов. Во-вторых, неоднозначность или омонимия, когда есть несколько способов разбить часть текста на токены. Эту проблему можно решить с помощью особых эвристик, таких как поиск самой длинной соответствующей цепочки из словаря (Longest Matching) и выбор варианта сегментации с наименьшим количеством токенов (Maximal Matching).

1) Longest Matching

   Это один из первых методов, использующихся для сегментации тайских слов (Poowarawan, 1986). В этом методе входная строка обрабатывается слева направо и граница проводится между самыми

длинными словами, найденными в словаре. В случае, если выбранный алгоритмом вариант не позволяет найти в словаре следующее слово, алгоритм возвращается на шаг назад и выбирает второе по длине слово и таким образом продолжает поиск границ. Понятно, что в этом случае алгоритму легко ошибиться в некоторых случаях из-за “жадности” (greedy characteristic). Например, словосочетание ไปหามเหสี (идти к королеве) алгоритм LM неверно разделит, как ไป (идти) + หาม (нести) + เห (отклониться) + สี (цвет). Правильный вариант: ไป (идти) + หา (к) + มเหสี (королева).

2) Maximal matching

Этот метод был предложен для решения проблемы предыдущего алгоритма (Sornlertlamvanich, 1993). Алгоритм генерирует все возможные варианты разделения предложения на слова, после чего выбирает тот вариант, в котором меньше слов. Реализуется это с помощью динамического программирования. Таким образом, алгоритм справляется с проблемой “жадности”, но возникает новая проблема. В случае, если находится несколько возможных разбиений текста на одинаковое количество слов, алгоритм не может определить лучший и приходится прибегать с некоторым эвристикам, например, снова использовать Longest Matching.

Примеры разделения словосочетания с помощью этого метода[4]:

1. ไปหามเหสี (идти к королеве)

* ไป (идти) + หาม (нести) + เห (отклониться) + สี (цвет) = 4 токена

[4] Примеры взяты из статьи *S. Meknavin, P. Charoenpornsawat and B. Kijsirikul*, 1997.

ไป (идти) + หา (к) + มเหสี (королева) = 3 токена

→ ไป (идти) + หา (к) + มเหสี (королева), т.е. выбран верный вариант

2. ตากลม (круглый глаз)

* ตาก (выставить) + ลม (ветер) = 2 слова

ตา (глаз) + กลม (круглый) = 2 слова

→ ตาก (выставить) + ลม (ветер), т.е. выбран неверный вариант.

3. ยานอก (импортное лекарство)

* ยาน (транспорт) + อก (грудь) = 2 слова

ยา (лекарство) + นอก (снаружи/вне) = 2 слова

→ ยาน (транспорт) + อก (грудь), т.е. выбран неверный вариант.

### *2.3.3. Методы машинного обучения*

Методы машинного обучения предназначены для решения проблем, возникающих при словарных методах. В них используется размеченный корпус, где явно обозначены границы между словами. С помощью машинного обучения можно построить статистическую модель на основе признаков (features) относящихся к символам на границах слов. Чаще всего для сегментации тайского текста к таким признакам относят классы букв, входящих в n-граммы потенциальных границ слов. Например, есть гласные, которые чаще всего встречаются в начале слова, в то время как диакритические знаки (знаки, указывающие на тон слога) никогда не начинают слог. Наиболее сложные алгоритмы используют также контекстную информацию, частеречную разметку (POS-теги), устойчивые словосочетания и семантику.

В машинном обучении сегментация слов формулируется как задача

бинарной классификации, где каждый символ из текстовой строки может принадлежать к одному из двух классов: начальный символ (B) и символ внутри слова (I). Таким образом тренировочные данные для машинного обучения могут выглядеть так:

| | | |
|---|---|---|
| ภ | C | B |
| า | V | I |
| ษ | C | I |
| า | V | I |
| ไ | V | B |
| ท | C | I |
| ย | C | I |

Таблица 1. Словосочетание ภาษาไทย (в первом столбце - буквы, во втором - C-согласные, V-гласные, в третьем - B-beginning, I-intra-word).

Также каждому символу приписывается признак на основании его n-грамма (от 3 до 11).

Основное преимущество методов машинного обучения в том, что им не требуются словари. Проблемы незнакомых слов и неоднозначности решаются путём извлечения достаточно большой контекстной информации из n-грамм и наличием большого набора тренировочных данных для точной классификации. Основной же недостаток этих методов в их сильной зависимости от источников текстов для тренировочного корпуса и его размера. Например, если построить модель, основанную на корпусе, состоящем из текстов лишь одного специфичного источника, то на текстах отличной тематики модель может показать низкие результаты.

#### *2.3.3.1. Метод, описанный в статье (S. Meknavin, P. Charoenpornsawat and B. Kijsirikul, 1997)*

Один из наиболее значимых факторов, влияющих на качество сегментации, это контекстная зависимость неоднозначных строк. Неоднозначность можно разделить на 2 типа:

1. Контекстно-независимая неоднозначность сегментации.

   Этот тип омонимии можно разрешить не используя широкий контекст. Несмотря на то, что существует несколько способов сегментации, есть лишь один приемлемый и правдоподобный способ разделить строку на слова, в то время как альтернативные варианты разбора имеют очень низкую вероятность. Однако простые алгоритмы LM и MM не всегда могут снять эту омонимию, зато вероятностные методы оказываются очень эффективными.

2. Контекстно-зависимая неоднозначность сегментации.

   В данном случае лишь контекст помогает определить, какой из вариантов разбора верный, так как все альтернативы потенциально и статистически возможны. Такая омонимия встречается реже, и тем не менее представляет большую трудность для алгоритмов сегментации.

Чтобы избавиться от неоднозначности, авторы статьи предлагают методы машинного обучения для разделения слов. В этих методах неоднозначность границ слов разрешается с помощью использования статистики частеречных n-грамм и частотности слов. Далее идет поиск

наиболее вероятной цепочки слов для данной строки/предложения. Такой метод более эффективен, чем Maximal Matching. Использование n-грамм слов может дать очень подробную информацию, но это требует огромного тренировочного корпуса и достаточно много компьютерных ресурсов для размещения таблиц n-грамм. Еще один недостаток n-грамм в том, что они не учитывают неупорядоченные устойчивые словосочетания (collocations), которые находятся на большом расстоянии друг от друга.

Для решения этих проблем авторы статьи предлагают метод, основанный на выделении признаков. Признаком может быть что-либо, указывающее на особенности контекста выбранного слова (слова слева, слова справа, устойчивые словосочетания).

На вход подается предложение $S$, получаем все возможные разборы предложения $S_1,S_2,...,S_n$, где $S_i = w_{i1}w_{i2}...w_{im}$. Задача состоит в том, чтобы определить из контекста какой из разборов предполагается. Для этого помимо N-граммов авторы используют два типа признаков: контекстные слова и коллокации. Признаки контекстных слов бинарны - они проверяют наличие или отсутствие определенных слов в пределах +/- K слов контекста анализируемого слова. Коллокации также бинарны - проверяется соответствие шаблонам слов вплоть до L прилегающих слов и/или POS-тегов. Чтобы автоматически извлекать отличительные признаки и использовать их для дизамбигуации, авторы статьи использовали два алгоритма: RIPPER и Winnow.

RIPPER:

Это обучающий алгоритм классификации по пропозициональным формулам (логическим правилам) вида: if $T_1$ and $T_2$and … $T_n$ then class $C_x$ ( $C_x$ - целевой класс, который необходимо изучить). Условия $T_i$ проверяют наличие или отсутствие того или иного признака, либо соответствие одной из четырех форм: $T_i = v$ (некоторое определенное значение), $T_i \geq v$, $T_i \leq v$, $T_i \subseteq V$ (принадлежность определенному множеству). Условие также может быть отрицательным. Алгоритм может воспринимать признаки, характеризуемые набором значений, что является большим преимуществом. Количество классов не ограничено (для мультиклассовой классификации алгоритм используется по правилу One-vs-all).

В процессе формирования логического правила алгоритм начинает поиск с пустым правилом, постепенно “отрезает” неподходящие для класса разборы путем добавления новых условий в правило до тех пор, пока все примеры не будут разделены на классы. Затем некоторая часть условий убирается, чтобы избежать “переобучения” алгоритма и достичь высокой эффективности на новых предложениях для разбора.

Winnow:

Алгоритм изначально описан в работе (Blum 1997). Это сеть из нескольких узлов (nodes). Каждый узел рассматривается как “специалист”, ответственный за определенное значение признака. На основе этого признака узел “голосует” за тот или иной класс для анализируемого примера. Далее методом взвешенного большинства определяется “победитель”. Также каждому узлу присваивается вес. Изначально значение всех весов равно единице, далее вес узла меняется в зависимости от правильности его голоса. Правильный голос увеличивает вес узла в 3/2 раза, а неправильный - делит

пополам. Примечательно, что узел-”специалист” может воздержаться от голосования, если в анализируемом примере не встретилось значения признака, который данный узел должен рассматривать. Основные преимущества алгоритма Winnow в том, что  он очень быстрый и не чувствительный к излишним признакам.

Описание эксперимента.

Для испытания обоих алгоритмов предлагается использовать 10 признаков-атрибутов: 4 признака для первого и второго слова перед и после целевого (анализируемого) слова, 4 признака для соответствующих POS-тегов и 2 атрибута для двух наборов по 10 слов перед и после целевого слова.

Кроме того, эксперимент был проведен на двух системах: FEATURE-1 и FEATURE-2.

- В системе FEATURE-1 составлен специальный словарь, где для каждого омонимичного случая записан свой набор вариантов сегментации (confusion set). В зависимости от признаков и контекста определяется наиболее вероятный вариант сегментации. Например, для строки “มากว่า” набор вариантов сегментации - C = {มา,กว่า,มาก,ว่า}. Для данного случая были выбраны 2 наиболее информативных признака: (1) มากว่า + число, (2) слово พูด в пределах -10 слов.
- В системе FEATURE-2 формируются всевозможные наборы префиксов из списка слов, для каждого из которых алгоритм обучается признакам. Набор префиксов - это такой набор слов, что если a и b являются элементами набора, то либо a - префикс к b, либо b - префикс к a. Пример такого набора {มา, มาก, มากมาย}

В эксперименте использовался корпус из 25,000 предложений, вручную разделенных на слова с расставленными тегами частей речи. В каждом предложении есть участок, который можно было бы разделить на слова несколькими способами. Корпус специально разделили на две части: предложения, содержащие контекстно-независимую неоднозначность и контекстно-зависимую. В каждом из них 80% - тренировочный набор и 20% - тестовый.

Точность измерялась не только на тестовом наборе, но и на тренировочном. Даже на тренировочном наборе редко достигается точность 100%. Результаты эксперимента показали, что оба алгоритма показывают более высокую точность и лучше справляются на примерах с контекстно-независимой неоднозначностью. Для сравнения также были протестированы алгоритм ММ и Триграм, точность которых еще ниже. Лучше всего с сегментацией справился алгоритм Winnow, реализованный с системой FEATURE-1. Эта система учитывает все неоднозначные случаи в корпусе, но на новых примерах, возможно, будут ошибки. Поэтому в будущей работе нужно либо учесть как можно больше таких случаев, либо попробовать скомбинировать две системы FEATURE-1 и FEATURE-2.

| | Context Independent | | Context Dependent | |
|---|---|---|---|---|
| | Training Set (%) | Test Set (%) | Training Set (%) | Test Set (%) |
| Maximal Matching | 79.74 | 78.85 | 52.10 | 53.52 |
| Trigram | 99.81 | 99.77 | 73.30 | 73.15 |
| FEATURE-1-RIPPER | 99.94 | 99.74 | 96.98 | 86.60 |
| FEATURE-1-Winnow | 100.00 | 99.70 | 100.00 | 95.33 |
| FEATURE-2-RIPPER | 98.52 | 91.27 | 93.28 | 89.00 |
| FEATURE-2-Winnow | 99.97 | 93.82 | 100.00 | 92.10 |

Таблица 2. Результаты работы алгоритмов ММ, Trigram, Ripper и Winnow на данных с наборами признаков FEATURE-1 и FEATURE-2.

### *2.3.3.2. Метод коллокаций*

В статье (Aroonmanakun, W, 2002) используется несколько иной метод, состоящий из двух этапов: деление текста на слоги с помощью статистики триграмм, а затем “склеивание” слогов в зависимости от их коллокаций.

Слог в тайском языке намного легче определить, чем слово, так как существует конечное число шаблонов, соответствующих тайским слогам. Поэтому автор считает, что надежнее будет не делить целую строку на слова, а наоборот, составлять слова из слогов.

Второй этап сегментации более сложный. Коллокации можно понимать, как меру внутренней связности сочетания слогов или слов. Высчитывается это как отношение вероятности встретить 2 слога вместе к вероятности встретить другой слог между имеющимися двумя. Предполагается, что если слово состоит из двух и более слогов, эти слоги очень часто будут встречаться вместе и вероятность их “встречи” намного выше, чем случайно. Кроме того, учитывается и то, что мера связности (коллокация) слогов на границах слова с пограничными им слогами из других слов должна быть минимальной. Поэтому при вычислении эту меру вычитают из меры внутренней связности слова. Получается следующая формула:

$$St = \sum_{i=1}^{n} F w_i - \sum_{i=1}^{n-1} D w_i, w_{i+1}$$

где первая сумма - внутренняя связность слова, а вторая - связность между слогами пограничных слов. Каждая граница между слогами имеет определенную вероятность быть границей слова. Таким образом входящее предложение алгоритм делит на слоги, а затем высчитывает меру внутренней связности между каждыми двумя слогами.

В результате эксперимента автор статьи получил F-меру 97.86% на тестовом корпусе, состоящем из 30,498 слогов (размер тренировочного корпуса - 553,372 слогов). Однако, на незнакомом тексте с незнакомыми словами система показала более низкий результат - 81.03%, что опять-таки свидетельствует о зависимости алгоритма от размера и качества корпуса.

#### *2.3.3.3. Метод машинного обучения C4.5*

В статье (Virach Sornlertlamvanich, Tanapong Potipiti and Thatsanee Charoenporn, 2000)_авторы предлагают алгоритм машинного обучения с учителем C4.5 для сегментации тайского текста на слова. Это алгоритм построения деревьев решений, применяемый для классификации объектов.

На первом этапе есть корень и множество объектов, которые необходимо разделить на подмножества. Далее в каждом шаге выбирается один из атрибутов, разбивающий множество на несколько ветвей. Атрибут выбирается по принципу наибольшего прироста информации, полученной при делении на подмножества. Ветвление продолжается, пока все объекты не будут классифицированы. Чтобы избежать переобучения алгоритма, некоторые ветви отсекаются. Для этого алгоритм рекурсивно проходит по всем подграфам дерева решений и проверяет, насколько изменится ожидаемая ошибка, если данный подграф изменить на лист (отсечь часть ветвей). Если после очередного ветвления в подграфе оказываются объекты одного класса, то подграф становится листом.

Алгоритм C4.5 реализован в программе Weka.

Признаки (атрибуты):

В качестве признаков для машинного обучения были выбраны: взаимная информация, энтропия, частота, длина строки и наличие служебных слов.

1) Взаимная информация слева и справа (Right and left mutual information).

   Взаимная информация случайных переменных (событий) a и b - это отношение вероятности, что a и b встретились вместе (co-occur), к произведению вероятностей независимых событий a и b. Формулы для левой и правой взаимной информации:

   $$Lm(xyz) = \frac{p(xyz)}{p(x)^* p(yz)} \qquad Rm(xyz) = \frac{p(xyz)}{p(xy)^* p(z)},$$

   где x - крайний слева символ, y - центральная подстрока, z - крайний сивол справа.

   Высокое значение взаимной информации означает, что совмещение a и b не случайно. То есть если “xyz” - это слово, то Lm(xyz) и Rm(xyz) будут высокие.

2) Левая и правая энтропия.

   Энтропия - это мера беспорядочности переменных.

   Формулы для левой и правой энтропии:

   $$Le(y) = -\sum_{\forall x \in A} p(xy|y)^* log_2 p(xy|y)$$

   $$Re(y) = -\sum_{\forall z \in A} p(yz|y)^* log_2 p(yz|y),$$

   где y - рассматриваемая строка, A - алфавит, x и z - любые буквы алфавита.

   Если y - это слово, то Le(y) и Re(y) будут иметь высокие значения.

3) Частота.

   Этот признак зависит от размера корпуса, поэтому его необходимо нормализовать путем деления частоты на размер корпуса и умножения на среднюю длину тайского слова:

$$F(s) = \frac{N(s)}{Sc} * Avl$$

,

   где s - рассматриваемая строка, N(s) - частота встречаемости слова s в корпусе, Sc - размер корпуса и Avl - средняя длина тайского слова.

4) Длина слова.

   Вероятность встретить случайную короткую строку больше, чем случайную длинную, поэтому при дизамбигуации строки разной длины нужно анализировать по-разному.

5) Служебные слова (Functional words).

   Такие частицы как จะ (буду) или ก็ (тогда) очень часто встречаются в тайских текстах. Поэтому чтобы отфильтровать эти частицы используется специальный атрибут Func(s) = 1 (если строка s содержит служебное слово) и Func(s) = 0 (если не содержит).

Используя немаркированный корпус размера 1 Mb, состоящий из 75 статей различных тематик, авторы извлекли все возможные строки длины от 2 до 30 знаков, встречающиеся в корпусе не менее 2 раз, имеющие положительную левую и правую энтропию. На этом этапе было получено 30,000 строк. Далее для обучения алгоритма и создания тренировочной и тестовой выборки все строки вручную маркируются специалистами как слова или “не-слова”.

В итоге, полученная точность составляла 87.3% на тренировочном корпусе и 84.1% на тестовом корпусе, полнота на обоих выборках - 56%. Авторы полагают, что результаты были бы намного лучше, если бы размер корпуса был больше.

#### *2.3.3.4. Метод условных случайных полей (CRF)*

Еще один эффективный метод машинного обучения описан в статье (C. Haruechaiyasak, S. Kongyoung, 2009). На основе этого метода была создана программа Tlex, которая сейчас в открытом доступе.

В этой статье описан алгоритм сегментации тайского текста на основе метода Условных Случайных Полей (Conditional Ramdom Fields). Условные случайные поля — это статистический алгоритм классификации, учитывающий контекст классифицируемого объекта. CRF является дискриминативной ненаправленной вероятностной графической моделью (Романенко А.А., 2014). Основное преимущество этого метода в том, что он не требует моделировать вероятностные зависимости между наблюдаемыми переменными. В отличие от марковской модели максимальной энтропии (MEMM), алгоритм CRF не имеет проблемы смещения метки (label bias problem).

Для машинного обучения были выбраны следующие наборы признаков:

1. Символ (каждый символ рассматривается как отдельный признак).
2. Тип символа (все символы были разделены на 10 типов. Подробнее - см. Таблицу 3).
3. Комбинированные признаки (используются символы и их типы).

| Тег | Описание | Значения |
|---|---|---|
| c | Согласные, которые могут находиться в конце слога | ก ค ข ฆ ง จ ช ซ ญ ฎ ฏ ฐ ฑ ฒ ณ ด ต ถ ท ธ น บ ป พ ฟ ภ ม ย ร ล ว ส ษ ศ ฬ อ |
| n | Согласные, которые не могут находиться в конце слога | ฉ ผ ฝ ต ห ฮ ฌ |
| v | Гласные, которые не могут начинать слог | ะ ิ ี ึ ื ั ุ ู ำ า ๅ |
| w | Гласные, которые могут начинать слог | เ แ ไ ใ โ |
| t | Тональные знаки (диакритики) | ่ ้ ๊ ๋ |
| s | Символьные знаки | ๆ ฯ ์ . |
| d | Цифры | 0-9 |
| q | Кавычки | “ ” ‘ ’ |
| p | Пробельные символы внутри слова | _ |
| o | Другие символы | a-z A-Z |

Таблица 3. Типы символов.

В ходе эксперимента корпус был преобразован в три различных тренировочных сета, размеченных в соответствии с тремя наборами признаков. По утверждению авторов, наилучший результат был достигнут при использовании комбинированных признаков. Полученная F1-мера составила 93.90%. Удивительно, что удалось добиться такого качества, не используя контекстной информации и n-грамм.

Также авторы предлагают все именованные сущности (NE - named entities), размеченные в корпусе и далее извлеченные, не разделять, если они

состоят из нескольких токенов (как, например, имя и фамилия персоны), а представлять как один токен. Это повышает F1-меру до 94.27%.

Метод был протестирован на текстах разных жанров и наилучшего качества удалось добиться на текстах разговорного стиля, газетных и энциклопедических статьях, рассказах и буддийских текстах. Наихудший результат (F1=85.80%) получился на королевских новостях. Дело в том, что королевская лексика сильно отличается от обычной, в ней очень много специальных слов. Поэтому неудивительно, что в данной тематике результаты оказались хуже всего.

Данный метод был также использован в статье (Canasai Kruengkrai, Virach Sornlertlamvanich, Hitoshi Isahara, 2006), однако авторы использовали не столь полезные признаки для машинного обучения и в результате метрики точности и полноты получились недостаточно высокими.

Метод условных случайных полей пользуется популярностью среди лингвистов и часто используется для сегментации текста на языках без пробелов, например, на японском (T. Kudo, K. Yamamoto, and Y. Matsumoto, 2004) и китайском (F. Peng, F. Feng, and A. McCallum, 2004) языках.

#### *2.3.3.5. Лексико-семантический метод*

Еще один интересный подход был описан в статье (K.Khankasikam, N.Muansuwan, 2005). Авторы добавили дополнительный семантический анализ для выбора правильной сегментации текста. Все слова разделены на 74 семантические категории AKO - “A Kind Of” (например, люди, организации, топографические объекты, события и т.п.). Для эксперимента был использован корпус ORCHID, в котором 431,338 слов, каждому из

которых была приписана категория АКО. Сегментация текста проходит в 4 этапа: генерация всех возможных вариантов сегментации (с помощью словарного метода ММ), анализ имен собственных (используя специальную базу имен), расстановка семантических тегов (категорий АКО) и семантический анализ. На последнем этапе комбинация получившихся тегов сравнивается с частотой такой комбинации в корпусе, и выбирается вариант с наибольший частотой.

Однако недостаток метода в том, что он зависит от полноты базы имён собственных и размера словаря с семантическими тегами. Если в сегментируемом предложении находится имя, которого нет в базе, то алгоритм ошибается. Поэтому на разных текстах с разным количеством имён собственных алгоритм показывал разную точность.

# Глава 3. Сравнительный анализ словарных методов и машинного обучения

В статье (C.Haruechaiyasak, S.Kongyoung and Matthew N.Dailey, 2008) авторы провели эксперимент, в ходе которого сравнили эффективность словарных методов и машинного обучения для сегментации текста на тайском языке.

Был использован корпус тайских текстов ORCHID (113,404 слов, размеченных вручную). Корпус был преобразован в n-граммы различных типов символов. Каждому символу был приписан признак ‘B’ или ‘I’ (как было выше сказано, ‘B’ - начало слова, ‘I’ - остальные буквы слова). Для оценки качества алгоритмов авторы использовали перекрестную проверку (10-fold cross validation). Для реализации алгоритмов Наивного Байесовского классификатора и Деревьев решений использовалась программа Weka, в которой эти алгоритмы доступны среди прочих. Для реализации метода опорных векторов (SVM) использовалась библиотека LIBSVM (в качестве функции kernel для тренировки модели авторы выбрали полином степени 3). И для метода условных случайных полей (CRF) использовалась программа CRF++, в которой реализован данный алгоритм.

В следующей таблице представлены результаты работы всех четырёх алгоритмов - точность, полнота и F1-мера. В качестве признаков были выбраны n-граммы размера 3, 5, 7, 9 и 11.

| | 3-gram | | | 5-gram | | | 7-gram | | | 9-gram | | | 11-gram | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| NB | 74.60 | 41.20 | **53.10** | 68.20 | 56.10 | **61.50** | 70.70 | 57.00 | **63.10** | 69.50 | 59.70 | **64.20** | 69.70 | 60.60 | **64.90** |
| J48 | 73.90 | 54.60 | **62.80** | 73.40 | 67.30 | **70.30** | 77.90 | 69.50 | **73.50** | 79.00 | 73.90 | **76.40** | 80.10 | 75.10 | **77.50** |
| LIBSVM | 75.91 | 52.59 | **62.14** | 73.48 | 67.45 | **70.33** | 80.11 | 72.88 | **76.32** | 86.49 | 76.93 | **81.43** | 92.87 | 88.71 | **90.74** |
| CRF++ | 89.92 | 87.28 | **88.58** | 95.05 | 93.40 | **94.22** | 95.59 | 94.63 | **95.10** | 95.61 | 94.84 | **95.23** | 95.79 | 94.98 | **95.38** |

Таблица 4. Результаты работы алгоритмов на различных n-граммных моделях (NB - Наивный Байес, J48 - Деревья решений).

Видно, что с увеличением размера n-грамма качество моделей немного улучшается, так как используется больше контекстной информации. В результате эксперимента наилучшие результаты показал метод условных случайных полей (CRF), хотя на 11-граммной модели метод опорных векторов справляется не намного хуже.

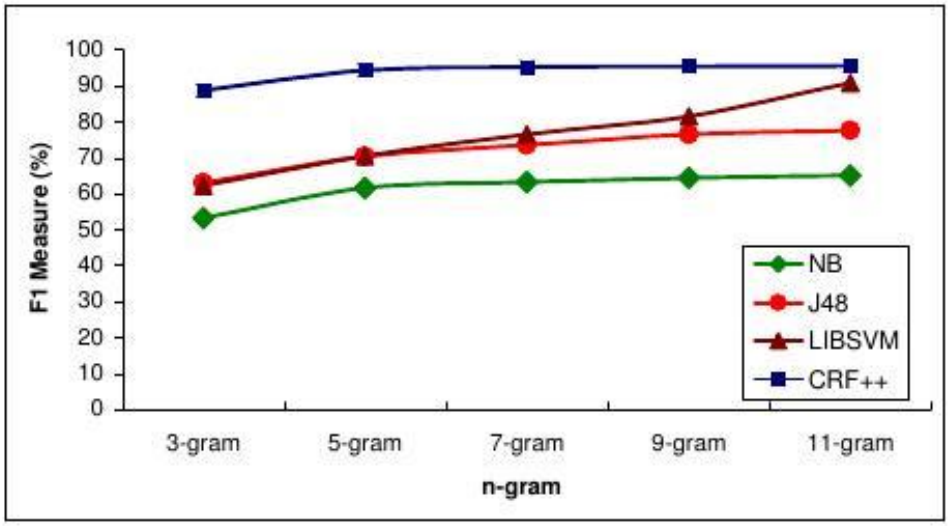


График 1. Сравнительный анализ работы алгоритмов машинного обучения на различных n-граммных моделях

Для реализации словарных методов использовалась программа SWATH[5] , в которой реализованы оба алгоритма Longest Matching (LM) и Maximal

[5] SWATH - открытый веб-ресурс для сегментации тайского текста. http://www.cs.cmu.edu/~paisarn/software.html

matching (MM). В качестве словарей использовался словарь Lexitron, в котором содержится примерно 30,000 слов, и набор слов из тренировочного корпуса для машинного обучения (Domain). В следующей таблице видны результаты обоих методов на обоих словарях в сравнении с алгоритмами машинного обучения. Видно, что MM совсем ненамного превосходит в качестве LM. Также интересно то, что использование словаря из корпуса улучшает качество обоих алгоритмов. В итоге, словарные методы оказались лучше Наивного Байеса, Деревьев решений и Метода опорных векторов, однако лучшее качество дал алгоритм CRF.

| Методы | Алгоритмы | Точность | Полнота | F1-мера |
|---|---|---|---|---|
| Словарные | LM-Lexitron | 88.21 | 86.91 | 87.55 |
| | LM-Domain | **95.20** | **88.55** | **91.75** |
| | MM-Lexitron | 88.34 | 87.39 | 87.86 |
| | MM-Domain | **95.27** | **88.92** | **91.98** |
| Машинное обучение | NB | 69.70 | 60.60 | 64.90 |
| | J48 | 80.10 | 75.10 | 77.50 |
| | SVM | 92.87 | 88.71 | 90.74 |
| | CRF | **95.79** | **94.98** | **95.38** |

Таблица 5. Результаты работы словарных методов с использованием двух словарей и алгоритмов машинного обучения.

## 3.1. Практическая часть

### *3.1.1. Тестирование инструментов токенизации тайского текста*

Так как в эксперименте, описанном выше, для оценки работы различных методов использовался достаточно устаревший корпус, обладающий рядом недостатков, я провела новое тестирование на корпусе InterBEST 2009, чтобы проверить, будут ли эти методы столь же эффективны на других текстах.

На данный момент есть несколько инструментов для сегментации тайского текста. Для тестирования были выбраны следующие программы:

1. Libthai[6]

   Программа представляет собой набор открытых (opensource) библиотек на языке C для обработки тайского языка. Сегментация текста осуществляется с помощью словарного метода MM. Словарь, используемый в программе, содержит 23,563 слова. Последние обновления в коде были в 2012 году.

2. Smart Word Analysis for THai : SWATH[7]

   Это программа с открытым кодом, разработана в 2003 году. Использует словарь Lexitron (30,000 слов) и реализует три алгоритма: LM, MM и модели биграмм частей речи.

3. Tlexs[8]

   Программа использует метод CRF для обучения модели сегментации[9].

[6] http://linux.thai.net/projects/libthai
[7] http://www.cs.cmu.edu/~paisarn/software.html
[8] http://sansarn.com/tlex/
[9] см. главу 3.4. Метод условных случайных полей

| **Название** (Метод) | **Libthai** (MM) | **SWATH** (LM, MM, bigram) | **Tlexs** (CRF) |
|---|---|---|---|
| Точность | 72.34 | 76.10 | **85.91** |
| Полнота | 65.08 | 66.07 | **86.99** |
| F1-мера | 68.52 | 70.73 | **86.44** |

Таблица 6. F1-мера протестированных программ тайской сегментации.

### *3.2.2. Результаты тестирования.*

Несмотря на то, что в результате вышеупомянутого исследования были достигнуты довольно высокие показатели для F1-меры, ни один из методов не дал результата выше 90.

Одна из причин в том, что авторы тестировали свои методы на корпусе ORCHID - небольшом корпусе, состоящем из текстов ограниченной и специфической тематики (служебных документов). Но на новом корпусе InterBEST’2009 результаты оказались хуже. Вторая причина в том, что составители корпуса InterBEST’2009 при разметке руководствовались принципом деления текста на наименьшие токены, то есть на максимально короткие токены при условии сохранения смысла. Поэтому многие слова, которые в словаре указаны, как один токен, в корпусе могут быть разбиты на несколько токенов. По той же причине Tlexs показал наилучший результат - алгоритм был обучен именно по этому корпусу.

## 3.2. Классификация ошибок токенизации

Качество работы токенизаторов принято оценивать путём подсчета ошибок, совершенных при неправильном делении текста. Как правило такие ошибки бывают двух видов: неверно поставленная граница и пропуск правильной границы. Это удобно при вычислении метрик точности и полноты.

Кроме этого, можно предложить и другую классификацию, более интересную при рассмотрении тайского языка. Ошибки токенизации тайского текста можно разделить на 3 типа:

1. Ошибки, препятствующие пониманию текста. При этом неправильно разделенные слова присутствуют в словаре. К таким ошибкам относится неправильное деление сильно связанных сложных слов, например, มือ (рука) + ถือ (держать) = мобильный телефон.
2. Ошибки, не препятствующие пониманию текста. Это касается неразличимости сложных слов и словосочетаний. В таких случаях сама необходимость деления на слова весьма спорна, как в примере คน (человек) + ขับ (водить) + รถ (машина) = водитель.[10]
3. Неправильная сегментация незнакомых слов. В основном, это заимствованные слова и неологизмы.

Я думаю, что все эти ошибки следует учитывать по-разному при измерении эффективности того или иного алгоритма токенизации. Ошибки

[10] подробней про это написано в разделе “Неоднозначность сегментации в тайском языке. Что считать границей слова?”

первого типа наиболее серьезны и действительно показывают, насколько хорошо обучен алгоритм. Такие ошибки не зависят от словаря. Ошибки второго типа мало значимы, так как практически не меняют смысл текста. В предыдущих исследованиях эти ошибки учитывались наравне со всеми, хотя даже для носителей языка не очевидно, стоит ли делить строку в таких случаях, или не стоит. Получается, что и тот, и другой вариант сегментации можно считать правильным. Что касается третьего типа, то такие ошибки зависят от словаря и поэтому их решить сложнее всего.

Если бы при оценке работы токенизаторов учитывались ошибки второго типа (а именно, вычитались из общего числа ошибок, как наименее значимые), то их показатели качества могли бы быть выше.

В качестве потенциального решения данной проблемы можно предложить создание нового корпуса или усовершенствование корпуса InterBEST’2009 таким образом, чтобы были размечены спорные варианты токенизации слов. То есть, если автор разметки не уверен, стоит ли делить данную строку на токены или нет, было бы полезно оставить и учитывать оба варианта сегментации, как правильные.

## 3.3. Эксперимент: реализация различных алгоритмов с помощью пакета Weka

Для тестирования использовался корпус InterBEST'2009, в котором обозначены только границы между словами. Все 509 размеченных документов разных тематик (5,036,228 слов) из корпуса были преобразованы в единый датасет в формате CSV следующего вида[11]:

| letter | next | previous | tag | mark |
|---|---|---|---|---|
| ธ | ◌ | . | c | B |
| ◌ | ด | ธ | v | I |
| ด | า | ◌ | c | I |
| า | แ | ด | v | I |
| แ | ห | า | w | B |
| ห | ' | แ | n | I |
| ' | ง | ห | t | I |
| ง | ไ | ' | c | I |

Таблица 7. Образец датасета для машинного обучения в Weka, преобразованного из корпуса InterBEST'2009.

В первом столбце - анализируемый символ. В качестве признаков использовались следующий за анализируемым символ, предыдущий символ, тип символа[12]. В пятом столбце - метка B (начало слова) и I (внутри слова), которую необходимо предсказать. Иначе говоря, алгоритм классификации должен разделить объекты (буквы) на 2 класса: B и I (позитивные и негативные).

С помощью программы Weka было протестировано несколько алгоритмов классификации. Так как датасет получился очень массивным, из

[11] см. код для преобразования корпуса в CSV и извлечения признаков в Приложении 1.
[12] см. таблицу 3.

него случайным образом были взяты 40,000 символов, из которых 10,128 принадлежат классу B и 29,872 - I. В таблицу результатов внесены алгоритмы, показавшие F1-меру выше 0.8:

| **Алгоритм** | **Точность** | **Полнота** | **F1-мера** |
|---|---|---|---|
| OneR<br>(minimum-error attribute) | 0.827 | 0.835 | 0.828 |
| RIpple DOwn Rule Learner(Ridor)<br>(rules method) | 0.864 | 0.85 | 0.855 |
| Winnow | 0.863 | 0.859 | 0.861 |
| NaiveBayes | 0.885 | 0.883 | 0.884 |
| Decision Table | 0.902 | 0.901 | 0.902 |
| DTNB<br>(decision table/naive bayes hybrid classifier) | **0.913** | **0.91** | **0.911** |
| J48<br>(decision trees) | **0.912** | **0.913** | **0.912** |
| PART<br>(Number of rules: 568) | **0.918** | **0.916** | **0.917** |
| IB1<br>(instance-based classifier) | **0.938** | **0.939** | **0.938** |

Таблица 8. Результаты классификации тайских букв с помощью алгоритмов пакета Weka (даны в порядке возрастания)

*Комментарии к результатам*

Несмотря на небольшой размер корпуса, результаты получились довольно высокими. Наилучшие показатели у алгоритма IB1, основанного на методе ближайшего соседа (класс определяется путем расчета нормализованного Евклидова расстояния между объектами классификации).

Были выбраны довольно простые признаки, которые, тем не менее, оказались наиболее релевантными для эффективного машинного обучения. Получается, даже с неполным корпусом и простыми признаками удалось получить F1-меру 0.938, что является хорошим результатом. С другой стороны, нужны более сложные признаки для того, чтобы улучшить полученную F1-меру.

Можно предположить, что если обучать модель на полном корпусе, то результаты получатся лучше.

Также я считаю, что можно сделать качественный инструмент для токенизации тайского текста, если попробовать совместить словарные методы и методы машинного обучения. То есть разделить токенизацию на два этапа: сначала найти все возможные варианты сегментации с помощью словаря, оставить однозначно сегментированные токены, а затем спорные варианты решить с помощью одного из методов машинного обучения (CRF или IB1), используя те же признаки, что в эксперименте.

## Выводы

В данной работе были исследованы особенности тайского языка и проблемы, с которыми приходится столкнуться при сегментации текста на слова и предложения, в частности неоднозначность сегментации и спорные случаи.

Был проведен подробный разбор существующих методов, выделены их сильные и слабые стороны. Также некоторые из методов и инструментов были протестированы на едином наборе текстов из корпуса InterBEST’2009. Сравнительный анализ показал, что наилучший результат дает метод условных случайных полей (CRF).

Помимо существующих и уже использованных методов были реализованы несколько новых методов в пакете Weka и получены достаточно высокие результаты. Признаки, выбранные для эксперимента с машинным обучением, оказались наиболее релевантными для определения границ слов в тайском языке.

Кроме того, были предложены несколько потенциальных путей улучшения существующих методов токенизации тайского текста, а именно: комбинация словарных методов и машинного обучения, совершенствование корпуса и учёт ошибок сегментации, допущенных в спорных случаях.

## Используемая литература:


1. *Морев Лев Николаевич,* Основы синтаксиса тайского языка, Наука, 1964.
2. Осипов Юрий Михайлович, Вопросы словообразования в современном тайском языке., М., 1969.
3. *Virach Sornlertlamvanich, Tanapong Potipiti, Chai Wutiwiwatchai and Pradit Mittrapiyanuruk,* The State of the Art in Thai Language Processing, ACL '00 Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000.
4. *Chaicharoen, N.* Computerized Integrated Word Segmentation And Part-Of-Specch Tagging Of Thai. Master Thesis, Faculty of Arts, Chulalongkorn University, 2002.
5. *Wirote Aroonmanakun,* Thoughts on Word and Sentence Segmentation in Thai, In Proceedings of the Seventh on Natural Language Processing, 2007.
6. *Yuen Poowarawan*, Dictionary-based Thai Syllable Separation, Proceedings of the Ninth Electronics Engineering Conference, 1986.
7. *Virach Sornlertlamvanich*, Word Segmentation for Thai, Machine Translation, 1993.
8. *A. Pornprasertkul,* Thai Syntactic Analysis. Ph.D. thesis, Asian Institute of Technology, 1994.
9. *A. Kawtrakul, C. Thumkanon, P. Varasarai and M. Suktarachan,* Automatic Thai Unknown Word Recognition. In Proceedings of Natural Language Processing Pacific Rim Symposium, 1997.

10. *S. Meknavin, P. Charoenpornsawat and B. Kijsirikul,* Feature-Based Thai Word Segmentation. In Proceedings of Natural Language Processing Pacific Rim Symposium, 1997.
11. *Virach Sornlertlamvanich, Tanapong Potipiti and Thatsanee Charoenporn,* Automatic Corpus-Based Thai Word Extraction with the C4.5 Learning Algorithm, COLING '00 Proceedings of the 18th conference on Computational linguistics, 2000.
12. *Theeramunkong, Usanavasin,* Non-Dictionary-Based Thai Word Segmentation Using Decision Trees, In Proceedings of the first international conference on Human language technology research, 2001.
13. *Aroonmanakun, W.*, Collocation and Thai Word Segmentation. In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop, Hua Hin, Thailand, 2002.
14. *Canasai Kruengkrai, Virach Sornlertlamvanich, Hitoshi Isahara*, A Conditional Random Field Framework for Thai Morphological Analysis, Proc. of the Fifth Int. Conf. on Language Resources and Evaluation (LREC-2006), 2006.
15. *Virach Sornlertlamvanich, Thatsanee Charoenporn, Hitoshi Isahara,* ORCHID: Thai Part-Of-Speech Tagged Corpus, The Journal of the Acoustical Society of Japan, 1999.
16. Segmentation guidelines for InterBEST 2009 Thai word segmentation: an international episode (Release 1, 1st June 2009).
17. *Chamyapornpong, S.*, A Thai Syllable Separation Algorithm. Master thesis, Asian Institute of Technology, 1983.

18. *Payothorn Urathamakun and Kanda Runapongsa,* Improved Rule-Based and New Dictionary for Thai Word Segmentation, The 3rd Joint Conference on Computer Science and Software Engineering, 2006.
19. *Choochart Haruechaiyasak and Sarawoot Kongyoun*g, TLex: Thai Lexeme Analyser Based on the Conditional Random Field, InterBEST 2009 workshop in SNLP 2009.
20. *Романенко, А.А.* Применение условных случайных полей в задачах обработки текстов на естественном языке, 2014
21. *T. Kudo, K. Yamamoto, and Y. Matsumoto,* "Applying Conditional Random Fields to Japanese Morphological Analysis," Proc. of EMNLP, 2004.
22. *F. Peng, F. Feng, and A. McCallum,* "Chinese segmentation and new word detection using conditional random fields," In Proc. of the 20th Int. Conf. on Computational Linguistics(COLING), 2004.
23. *Krisda Khankasikam, Nuttanart Muansuwan*, Thai Word Segmentation a Lexical Semantic Approach, Proceedings of the 10th Machine Translation Summit, 2005.
24. *Choochart Haruechaiyasak, Sarawoot Kongyoung and Matthew N. Dailey*, A Comparative Study on Thai Word Segmentation Approaches, Proc. of the ECTI-CON, 2008.

# Приложение 1.

Код на языке программирования Python для преобразования корпуса InterBEST’2009 в CSV-датасет и извлечения признаков для машинного обучения.

```
#-*- coding: utf-8 -*-

import re, os, glob
from pathlib import Path
folder = os.getcwd()
p = Path('.')

# create a new CSV-file for dataset
outfile = open('Thai_data.csv', 'w', encoding='utf-8')
outfile.write('id'+'\t'+'l'+'\t'+'next'+'\t'+'prev'+'\t'+'tag'+'\t'+
                'mark'+'\n')

for file in list(p.glob('*.txt')):
    with open(str(file), encoding='utf-8') as file:

        # cleaning data
        text = file.read()
        text = re.sub('<NE>', '', text)
        text = re.sub('</NE>', '', text)
        text = re.sub('<AB>', '', text)
        text = re.sub('</AB>', '', text)
        text = re.sub('| ', '', text)
        text = re.sub('\n', '', text)

        for i in range(len(text)-1):
            if str(text[i]) != '|':
                mark = 'I'
                next_l = '.'
                prev_l = '.'
                tag = 'o'

                # assigning a tag for symbol type
                if text[i] in ['ก','ค','ข','ฃ','ฆ','ง','จ','ช','ซ','
ญ','ฎ','ฏ','ฐ','ฑ','ฒ','ณ','ด','ต','ถ','ท','ธ','น','บ','ป','พ','ฟ',
'ภ','ม','ย','ร','ล','ว','ส','ษ','ศ','ฬ','อ']:
                    tag = 'c'
                if text[i] in ['ฉ','ผ','ฝ','ศ','ห','ฮ','ฌ']:
                    tag = 'n'
                if text[i] in ['ะ','ิ','ี','ึ','ื','ั','ุ','ู','า','ำ','ๅ']:
                    tag = 'v'
                if text[i] in ['เ','แ','ไ','ใ','โ']:
```

```
                    tag = 'w'
                if text[i] in ['่','้','๊','๋']:
                    tag = 't'
                if text[i] in ['ๆ','ฯ','์','.']:
                    tag = 's'
                if text[i] in
['0','1','2','3','4','5','6','7','8','9','0']:
                    tag = 'd'
                if text[i] in ['‘','’','‘','’']:
                    tag = 'q'
                if text[i] == '_':
                    tag = 'p'

                # next symbol attribute
                if i==(len(text)-1):
                    next_l = '.'
                    mark = 'I'
                if str(text[i+1]) != '|':
                    next_l = str(text[i+1])
                if i != (len(text)-2) and str(text[i+1]) == '|':
                    next_l = str(text[i+2])

                # previous symbol attribute
                if str(text[i-1]) != '|':
                    prev_l = str(text[i-1])
                if str(text[i-1]) == '|':
                    mark = 'B'
                    prev_l = str(text[i-2])
                if i==0:
                    prev_l = '.'
                    mark = 'B'


outfile.write('"'+str(text[i])+'"'+'\t'+'"'+next_l+'"'+'\t'+'"'
                +prev_l+'"'+'\t'+tag+'\t'+mark+'\n')
        file.close()
outfile.close()
```