Автоматический поиск и классификация однословных терминов в корпусе предметной области с использованием логарифмической меры сходства с неспециализированным корпусом

Automatic detection and classification of single-word terms in a specific domain corpus using log-likelihood similarity with general purpose corpus*

Гельбух А. Ф. (www.gelbukh.com),
Сидоров Г. О. (www.cic.ipn.m/~sidorov),
Лавин-Вийа Э.

Лаборатория естественного языка и обработки текста,
Центр компьютерных исследований (CIC),
Национальный политехнический институт (IPN), г. Мехико, Мексика

Чанона-Эрнандес Л.

Инженерный факультет (ESIME),
Национальный политехнический институт (IPN), г. Мехико, Мексика

В статье представлен метод поиска однословных терминов в корпусе предметной области, использующий логарифмическую меру сходства (log-likelihood) с неспециализированным корпусом. Также проводится автоматическая классификация полученных терминов на основе меры сходства по косинусу угла.

1. Введение

Автоматическое построение онтологий для специфических предметных областей, или, по крайней мере, построение списка терминов и гипотез о возможных отношениях между ними для последующей ручной обработки, является важной и актуальной задачей современной компьютерной лингвистики. Это связано с трудностями ручной разработки онтологий: необходимость в экспертах для каждой предметной области, невысокая скорость их работы, большие затраты, а также субъективность полученных данных (см. например, Uschold & Grunninger, 1996). Эта область является достаточно активно развивающейся. В настоящее время не существует какого-либо общепринятого метода построения онтологий.

В качестве первого шага такого построения является полезным извлечение однословных терминов из текстов или слов, которые являются частью многословных терминов. В дальнейшем из этих однословных терминов могут быть сформированы многословные термины. Заметим, что большинство терминов состоит из нескольких слов, и большинство существующих методов сразу пытается их извлекать, измеряя степень совместной встречаемости составляющих их слов. Казалось бы, может быть проще набросать проект выделения таких терминов вручную вместо разработки программы. Мы уже упоминали субъективность такого подхода. Кроме того, для одной-двух предметных областей это может быть проще, то если их уже сто или тысяча, то наверное проще применять программу автоматической обработки.

* Work done under partial support of Mexican Government (CONACYT, SNI) and National Polytechnic Institute, Mexico (SIP, COFAA, PIFI; projects SIP 20091587, 20090772, 20100773, 20100668).

Dialog'2010.indb 82 11.05.2010 16:56:59

В данной статье мы представляем метод, являющийся модификацией метода описанного для китайского языка в (Tingting He *et al*., 2006), и эксперименты построения проекта онтологии на материале испанского языка. Мы работали с предметной областью «Информатика». Метод является достаточно универсальным и может быть применен к широкому классу языков (в том числе, к русскому) с соответствующими изменениями в предобработке. Метод дает достаточно хорошие предварительные результаты, выделяя однословные термины предметной области и объединяя их в классы.

Далее в статье мы сначала описываем предлагаемый метод (предобработка, поиск терминов, мера их сходства и классификация). Затем приводятся данные о проведенном эксперименте, и в заключение делаются выводы.

2. Предлагаемый метод

Входными данными метода является корпус специфической предметной области (мы работали с текстами по информатике). Также метод предполагает наличие неспециализированного корпуса для его использования при сравнении. Заметим, что размер этих двух корпусов может быть очень разным.

Предлагаемый метод состоит из четырех основных шагов: предобработка и подготовка данных, поиск терминов, вычисление меры их сходства и объединение терминов в классы.

Метод использует традиционную в информационном поиске векторную модель представления набора документов, когда документы и слова из них являются двумя измерениями матрицы, которая содержит частоты данного слова в данном документе. Заметим, что эквивалентным является представление в виде таблицы содержащей набор: *слово-документ-частота*, например,

Табл. 1. Представление данных в виде таблицы

Слово	**Документ**	**Частота**
software	14	3
software	16	3
software	20	12
и т. п.	…	…

Очевидно, что в этом случае нет необходимости добавлять записи для документов, в которых данного слова просто нет, т. е. в которых слово имеет частоту 0. Таких записей просто не будет в этой таблице.

Метод был модифицирован в следующих аспектах по сравнению с описанным в (Tingting He *et al*., 2006) методом для китайского языка: он применяется к индоевропейскому языку (испанскому) с соответствующим изменением предобработки; мы не используем обогащение словаря дополнительными ресурсами (типа WordNet), что является существенной частью исходного метода, при этом метод продолжает оставаться достаточно надежным; мы изменили формулу вычисления разницы между корпусами — вместо вычисления разницы по алгоритму *loglikelihood test*, мы вычисляем разницу по другому алгоритму, тоже основанному на *loglikelihood*, который вычисляет «расстояние» между элементами (Rayson *et al*., 2004; Dunning, 1993). Заметим, что этот алгоритм как раз и предназначен для вычисления сходства корпусов.

2.1. Предобработка и подготовка данных

На этапе предобработки в документах выделяются слова. В нашем случае мы игнорировали знаки препинания, специальные символы, числа. Все слова приводятся в один регистр.

Все слова лемматизируются. Мы пользовались лемматизатором для испанского языка, разработанным в нашей лаборатории. Для русского языка также существуют доступные лемматизаторы (см. например, Gelbukh and Sidorov, 2005).

Кроме того, мы отфильтровываем все служебные слова (предлоги, союзы, вспомогательные глаголы, и пр.), так как заранее известно, что они не являются терминами.

Для полученных лемм подсчитываются их частоты в каждом документе и заносятся в матрицу.

Эта процедура проделывается отдельно для каждого корпуса. В результате мы получаем два матрицы (или таблицы), которые представляют весь корпус исходных текстов и весь неспециализированный корпус.

2.2. Поиск терминов с использованием логарифмической меры сходства

Как уже было сказано, метод использует два корпуса: корпус специфической предметной области и неспециализированный корпус. Основная идея состоит в сравнении взвешенных частот слов в двух корпусах, и если какое-либо слово гораздо чаще присутствует в корпусе предметной области, то это вероятный термин.

Заметим, что в (Tingting He *et al*., 2006) указано, что логарифмическая мера сходства дает лучшие результаты чем гораздо более традиционная мера TF/IDF.

Мы также применили логарифмическую меру сходства, но не в варианте теста (*loglikelihood test*, см. www.wikipedia.org), а в варианте, который предназначен для сравнения корпусов (Rayson *et al*., 2004).

Для каждого слова, вычисление проводилось по следующей формуле:

$$G = 2 * \left(\left(fr_{domain} * \log\left(\frac{fr_{domain}}{frExpected_{domain}} \right) \right) + \left(fr_{general} * \log\left(\frac{fr_{general}}{frExpected_{general}} \right) \right) \right)$$

где:

$frExpected_{domain}$ и $frExpected_{general}$ ожидаемые частоты в корпусе предметной области и в неспециализированном корпусе соответственно;

fr_{domain} и $fr_{general}$. реально наблюдаемые частоты в корпусе предметной области и в неспециализированном корпусе соответственно.

Табл. 2. Пример вычислений веса терминов

Слово	fr_{domain}	$fr_{general}$	$frExpected_{domain}$	$frExpected_{general}$	**G**
socket	1	0	0,010286744	0,989713252	9,153798
sofisticado (сложный)	5	169	1,789893508	172,210113500	3,912798
soft	1	12	0,13372767	12,866271970	2,351035
software	430	831	12,97158432	1248,028442000	2334,961
*sofware***	2	2	0,041146975	3,958853006	12,8037
sol (солнце)	2	933	9,618105888	925,381897000	−9,016687
solamente (только)	20	1714	17,83721352	1716,162842000	0,254846

** Это слово написано с ошибкой в корпусе. Правильно *software*

Для вычисления ожидаемых частот используются следующие формулы.

Обозначим через *R_fr* такое отношение частот:

$$R\_fr = \frac{fr_{domain} + fr_{general}}{size_{domain} + size_{general}}$$

где $size_{domain}$ и $size_{general}$ размеры соответствующих корпусов, вычисленные в количестве слов. Тогда

$$frExpected_{domain} = size_{domain} * R\_fr$$

$$frExpected_{general} = size_{general} * R\_fr$$

Следующий важный шаг в нашем алгоритме поиска терминов состоит в следующем. Заметим, что значения полученные по формуле, не различают, к какому корпусу относится предполагаемый термин (т. е. формула симметрична относительна корпусов). Нас это не устраивает, потому что мы ищем термины в предметной области, а не в неспециализированном корпусе. Для принятия во внимание этого явления, вычислим дополнительно относительные частоты слов в каждом корпусе и будем рассматривать только те слова, у которых больше относительная частота в корпусе предметной области. Для этого, если относительная частота в корпусе предметной области меньше, чем в в неспециализированном корпусе, например, умножим результат, полученный по вышеуказанной формуле, на −1.

В Таблице 2 приведены примеры полученных вычислений.

Например, у слова *software* (*программное обеспечение*) получился очень высокий вес. Вес же слова *sol* (*солнце*), хотя и достаточно высок, но был умножен на −1, так как его относительная частота больше в неспециализированном корпусе.

По окончании этого этапа метода, у нас есть список слов с их весом в корпусе предметной области, который соответствует их вероятности быть терминами этой области. Остается открытым вопрос о том, по какому порогу провести границу между терминами и не-терминами (см. раздел Эксперименты).

2.3. Вычисление меры сходства терминов по косинусу угла

Следующий этап метода состоит в вычисление меры сходства терминов по косинусу угла (см. например, cosine similarity в wikipedia). Эта мера будет использоваться для классификации терминов на следующем шаге. В данном вычислении используется стандартная формула из информационного поиска. В качестве данных мы, как это обычно делается, используем частоты отобранных слов. Обычно эта мера сходства отражает совместную встречаемость слов в одном документе, нам кажется, что эта интерпретация применима и в нашем случае.

Естественно, вычисления проводятся только для слов, отобранных на предыдущем этапе алгоритма.

$$\cos(x,y) = \frac{A \cdot B}{|A||B|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\ \sqrt{\sum_{i=1}^{n} y_i^2}}$$

где:

n это количество документов в корпусе предметной области;

x_i и y_i это частота слов x и y в документе i.

В Таблице 3 приведены примеры вычислений сходства разных слов со словом *software (программное обеспечение)* в алфавитном порядке.

Табл. 3. Пример вычислений сходства терминов для слова *software*

Слово	Сходство
algoritmo (алгоритм)	0,032
almacén (склад)	0,018
almacenamiento (хранение)	0,044
almacenar (хранить)	0,086
aplicación (приложение)	0,420
archivo (файл)	0,203
arpanet (arpanet)	0,031
arquitectura (архитектура)	0,271
artificial (искусственный)	0,029
base (основа)	0,119
bioinformática (биоинформатика)	0,073
cálculo (вычисление)	0,055
característica (характеристика)	0,220
ciencia (наука)	0,071
circuito (плата)	0,018
código (код)	0,759
compilador (компилятор)	0,185
componente (компонента)	0,171
и т. д.	...

Можно заметить, что наибольшее сходство у слова *software* среди слов, отобранных как потенциальные термины в области информатики, со словом *код* (0,759) и наименьшее со словом *биоинформатика* (0,073).

2.4. Классификация терминов

В качестве алгоритма классификации мы использовали стандартный алгоритм *k-means* (www.wikipedia.org).

Этот алгоритм получает в качестве входного параметра количество классов k, которые должны быть сформированы. Алгоритм состоит в следующем:

- Случайным образом выбираются k терминов в качестве центров классов.
- Каждый оставшийся термин классифицируется на основе сходства с центром класса (по наибольшему сходству, см. выше).
- Заново высчитываются центры классов.
- Два предыдущих шага повторяются до тех пор, пока есть какие-то изменения в результатах.

В результате получаются k классов, состоящих из наболее похожих терминов. См. пример в Таблице 4.

Табл. 4. Пример результатов классификации в одном из экспериментов

Центр	Элементы класса	Сходство с центром
describir (описать)	*describir (описать)*	1,000
	solucionar (решить),	0,729
	arquitectura (архитектура компьютера),	0,530
	matemática (математика),	0,509
	interfaz (интерфейс),	0,449
	patrón (образ, шаблон),	0,439
	diseñar (разрабатывать),	0,427
	diseñador (разработчик)	0,335
disco (диск)	*disco (диск),*	1,000
	dvd (DVD),	0,947
	disquete (дискета),	0,934
	rom (ROM),	0,933
	cd (CD),	0,928
	usb (USB),	0,899
	flash (флеш),	0,877
	almacenamiento (хранение),	0,872
	óptico (оптический),	0,840
	velocidad (скорость),	0,753
	soportar (поддерживать)	0,591
и т. п.	...	...

Эксперименты

2.5. Данные и параметры

В наших экспериментах мы использовали следующие данные (на испанском языке). Для сравнения в качестве неспециализированного корпуса были выбраны выпуски газеты Excélsior (Мексика) конца 90-х годов, всего 1 365 991 слов.

В качестве корпуса предметной области были взяты страницы из *wikipedia*, связанные с информатикой: *информатика, программное обеспечение (software), программирование*, и т. п. Всего было загружено 26 страниц, содержащих 44 495 слов. В принципе, для экспериментов можно взять любую коллекцию текстов и это не требует какого-либо дополнительного обоснования.

После нескольких предварительных экспериментов, мы выбрали порог для алгоритма поиска терминов в 270 терминов. В статье (Tingting He *et al.*, 2006) был выбран порог в 216 терминов, при том, что дополнительно производился анализ отношений в китайском аналоге WordNet. В качестве порога на количества классов, мы выбрали порог в 19 классов, для

сравнения в указанной статье был выбран порог в 20 классов. Эти параметры легко варьировать в экспериментах. Порог был выбран эмпирически, т. е., мы пробовали разные значения и остановились на пороге, который дает интуитивно лучшую классификацию.

2.6. Результаты

После применения нашего метода к указанным корпусами с данными параметрами, были получены следующие результаты. Приведем несколько терминов, которые получили наибольший вес в соответствии с алгоритмом поиска терминов, см. Таблицу 5.

Табл. 5. Термины области «Информатика» с наибольшим весом

Термин	**Log-likelihood**
Dato (данные)	1506
Computador (компьютер)	863
Circuito (плата)	467
Memoria (память)	384
Señal (сигнал)	372
Secuencia (последовательность)	353
Computación (вычисление)	351
Información (информация)	346
Dispositivo (устройство)	342
Algoritmo (алгоритм)	341
Electrónico (электронный)	322
Base (основа)	319
Diseñar (разрабатывать)	307
Utilizar (использовать)	282
и т. д.	...

Интересный вопрос состоит в том, нужно ли включать в этот список глаголы, т. к. большинство глаголов в этом случае являются просто лексическими функциями к соответствующим существительным. В случае если было бы принято решение об исключении глаголов, это легко сделать, т. к. был проведен морфологический анализ и лемматизация.

После применения алгоритма классификации к полученным терминам с порогом на 19 классов, были получены следующие результаты, см. Таблицу 6. Для простоты не будем приводить данные о сходстве слов с центром класса. Первое слово в каждом классе является его центром. Как можно заметить, некоторые слова приводятся по-английски (*for, to, DAQ*, и т. д.), как они были представлены в исходных текстах.

В таблице мы зачеркнули слова, которые явно НЕ являются терминами с нашей точки зрения и подчеркнули глаголы, которые мы не будем использовать для подсчетов, т. к. необходимость их включения в онтологию небесспорна. Оставляем открытым вопрос о том, нужно ли исключить также и отглагольные существительные.

Среди отобранных слов присутсвует большое количество слов общенаучной лексики, типа *анализ, система, модель, наука, теория,* и пр. Строго говоря, они не должны присутствовать в онтологии выбранной предметной области, но с другой стороны они все-таки являются научными терминами. Вероятно, их можно отфильтровать проделав аналогичное описанному сравнение корпусов, но уже из двух различных предметных областей. Мы выделили эти слова в Табл. 6 курсивом. Как обычно, четкого критерия выделения таких слов нет, поэтому мы руководствовались следующим принципом: «если у слова есть в данной области какое-то специфическое значение, отличное от общенаучного, то он считается термином», например, *объект* может быть термином в программировании (*объектно-ориентированное*; по-испански, *ориентированное на объекты*). Или слово *протокол*, когда существуют *протоколы передачи данных*, и пр.

Некоторые слова оказались в двух классах, потому что их расстояние до двух центров было одинаковым. Кроме того, необходимо пояснить, что когда система морфологического анализа не имела в в словаре какого-либо слова, она считала каждую его форму отдельной леммой. Как видно в таблице, такие слова обычно попадают в один и тот же класс, например, *genoma* и *genomas* (*геном* и *геномы*), что говорит о правильной работе алгоритма классификации.

Также можно наблюдать «случайное размазывание» по разным классам массива наиболее общих терминов предметной области (*программа, компьютер, пользователь, файл, интерфейс*), что, вероятно, связано с их присутствием во многих текстах коллекции.

Как обычно бывает в случае автоматических методов обработки, результаты очень редко стопроцентно точны, скажем, в разные классы попали слова *информация* и *информационный*, *параллельный* и *параллелизм*, и пр.

Представленные термины относятся к разным областям информатики: биоинформатике, электронике, программированию, и пр. Заметим, что в нашей коллекции были тексты, скажем, из биоинформатики, поэтому и были термины из области биоинформатики. Это зависит от выбора текстовой коллекции, из которой извлекаются термины (Таблица 6).

Как обычно в случае онтологий, оценить полученные результаты непросто, потому что не существует «золотого стандарта» для оценки. Кроме того, ручное составление онтологии процесс очень субъективный, в этом смысле не очень ясно, может ли такой стандарт существовать. Особенно трудно оценить полноту (recall), т. е. насколько в построенной онтологии не хватает каких-либо терминов.

Наверное, если речь идет об оценке, то можно сравнивать с каким-либо словарем, это дало бы точность и полноту, в каком-то приближении, скажем, в нашем случае, со словарем по информатике. По-испански, у нас не было доступного словаря. Кроме того, это не снимает вопроса о субъективности построения такого словаря, используемого для оценки.

Dialog'2010.indb 86 11.05.2010 16:56:59

В статье (Tingting He *et al*., 2006) после ручной оценки были получены результаты с точностью в районе 70 %. Повторимся, что оценка ручная, потому что нет стандарта для оценки.

Если мы оценим точность выделения терминов нашим методом, то у нас получатся следующие результаты. Всего терминов 270, из них 31 глагол (подчеркнуты), то есть остается 239 терминов. Из этих терминов 19 явно не являются терминами данной области (зачеркнуты). Считая таким образом, мы получаем точность в 92,5 %. Если мы добавим к словам, которые будем считать неправильно определенными, 48 общенаучных терминов (курсив), то получим точность в [239 − (19 + 48)] / 239 = 72 %.

Табл. 6. Классы полученные в результате работы алгоритма

algoritmo, for, *implementación*, array, implementar, árbol алгоритм, for, *реализация*, массив, реализовать, дерево (поиска)
analógica, voltaje, binario аналоговый, напряжение, бинарный
as, if, int, integer, pseudocódigo, return, vtemp, *diagrama*, *descripción*, Turing, end as, if, int, integer (число), псевдокод, return (возврат), vtemp, *схема*, *описание*, Тьюринг, end (конец)
b2b, *business*, hosting, cliente, servidor, internet, ~~to~~, electrónico, consistir B2B, *бизнес*, хостинг, клиент, сервер, Интернет, ~~to~~, электронный, состоять
biología, bioinformática, adn, alineamiento, clustalw, fago, gen, genoma, genomas, genome, genómica, génica, homología, *human*, microarrays, *modelado*, nucleótidos, *predicción*, proteína, proteína-proteína, sanger, secuenciación, evolutivo, *secuencia*, *biológico*, computacional, protocolo, *variedad*, *análisis, técnica, estructura, interacción*, completar, *montaje, herramienta,* ~~menudo~~, usar, talar, software, visualizar, *cuantificación*, *modelo*, automatizar, búsqueda *биология*, биоинформатика, ДНК, выравнивание, ClustalW, фаг, ген, геном, геномы, геном, геномика, генный, гомология, *человек*, микрочип, *моделирование*, нуклеотиды, *прогнозирование*, белок, белок-белок, Sanger последовательность, эволюционный, *последовательность*, *биологический*, вычислительный, протокол, *отбор, анализ, технологический, структура, взаимодействие,* дополнить, *монтаж, инструмент,* ~~часто~~, ~~использовать~~, ~~рубить~~, программное обеспечение, визуализировать, *количественная оценка*, *модель*, автоматизировать, поиск
componente, transistor, tubo, funcionar, conexión, dispositivo, ~~etc~~, *tecnología*, digitales, microprocesadores, *velocidad, lógica*, soler, altavoz компонента, транзистор, трубка, функционировать, связь, устройство, ~~и т. д.~~, *технология*, цифровые, микропроцессоры, *скорость, логика*, случаться, динамик
computación, *ciencia, constable, científica,* cómputo, *disciplina, matemática*, ~~usualmente~~, *teoría*, computacionales, *ingeniería*, estudiar, artificial, *matemático*, informática, paralelo, programación компьютер, *наука, стабильный, научный,* вычисление*, дисциплина, математика*, ~~как правило~~, *теория*, вычислительные, *инженерный*, исследовать, искусственный, *математический*, информатика, параллельный, программирование
conjunto, *notación, problema*, finito, binaria, complejidad, np, np-completo, *número, tamaño, elemento*, coste, lineal, ~~comúnmente~~, ~~montículo~~ множество, *обозначение*, *проблема*, конечный, бинарный, сложность, NP, NP-полный, *количество, размер, элемент*, стоимость, линейный, ~~обычно~~, ~~курган~~
código, compilador, compiladores, lenguaje, máquina, programa, ~~compuesto~~ код, компилятор, компиляторы, язык, машина, программа, ~~состоящий~~
descifrar, criptografía, cifrar, *método*, texto, denominar декодировать, криптография, шифровать, *метод*, текст, обозначать
dimensión, cubo, *espacial*, almacén, marts, metadato, middleware, warehouse, data, olap, tabla, operacional, variable, *definición*, especificar, usuario, poseer, almacenar, dato, colección, arquitectura, registro *измерение*, куб, *пространственный*, хранилище (данных), marts, метаданные, промежуточное программное обеспечение, хранилище (данных), данные, OLAP, таблица, оперативный, переменная, *определение*, указать, пользователь, иметь, хранить, данные, набор, архитектура (компьютера), запись
diseñar, diseñador, objeto, funcional, procesar, proceso *разработка, разработчик*, объект, функциональный, обработать, процесс
formato, avi, compresión, *especificación*, formatos, mov, archivar, vídeo, audio, archivo, informático, codificar, *estándar* формат, AVI, сжатие, *спецификация*, форматы, MOV, архивировать, видео, аудио файл, информационный, шифровать, *стандартный*

potencia, válvula, analógicos, semiconductor, corriente, alternar, analizador, electrónica, conmutación, eléctrico, sonido, pila, supercomputadoras напряжение, клапан, аналоговые, полупроводниковый, ток, изменять (полярность), анализатор, электронный, коммутация, электрический, звук, аккумулятор, суперкомпьютеры
red, *principal*, *artículo*, permitir, utilizar, ~~vario~~, aplicación, información, ~~través~~, ~~tipo~~, *sistema*, ~~ejemplo~~, característica, interfaz, *forma*, gestión, operativo, acceder, ~~diferente~~, base, contener, operación, función, clasificar, ordenador, ejecutar, programador, cálculo, modelar, relacionales, interfaces, objeto, relacional сеть, *основной, статья,* позволять, использовать, ~~различный~~, приложение, информация, ~~посредством~~, ~~тип~~, *система*, ~~пример~~, характеристика, интерфейс, *форма*, управление, оперативный, обратиться (к данным), ~~разный~~, база, содержать, операция, функция, классифицировать, компьютер, исполнять (программу), программист, расчет, моделировать, реляционные, интерфейсы, объект, реляционный
~~rápido~~, acceso, ~~sencillo~~, soportar, web, ~~específico~~, ~~central~~, fiabilidad, paralelismo ~~быстрый~~, доступ, ~~простой~~, поддерживать, Web, ~~конкретный~~, ~~центральный~~, надежность, параллелизм
señal, transductores, transductor, impedancia, filtrar, conversión, acondicionamiento, convertidor, daq, *adquisición*, analógico, conectar, adaptación, frecuencia, medir, tensión, sensores, digital, cable, control, ~~física~~, entrada, medición, ~~físico~~, salida, ~~normalmente~~, bus, dato сигнал, датчики, датчик, сопротивление, фильтровать, преобразование, упаковка, конвертер, DAQ, *приобретение*, аналоговый, соединять, адаптация, частота, измерять, напряжение, датчики, цифровой, кабель, управление, ~~физика~~, ввод, измерение, ~~физический~~, выход, ~~как правило~~, шина (данных), данные
térmico, ci, cápsula, integration, scale, chip, circuito, chips, integrar, híbrido, silicio, reproductor, amplificador, ~~fabricación~~ тепловой, CI, капсула, интеграция, масштаб, микросхема, плата, микросхемы, комплексный, гибридный, кремний, проигрыватель, усилитель, ~~производство~~

3. Выводы

В данной статье мы представили метод, который позволяет построить проект онтологии предметной области состоящий их однословных терминов, используя тексты данной области и неспециализированный корпус. В дальнейшем эти термины можно объединять в многословные. Метод позволяет определить слова, которые являются возможными терминами (или частями многословного термина) данной области используя логарифмическую меру сходства. После этого дополнительно термины классифицируются на основе меры сходства по косинусу угла с использованием алгоритма классификации *k-means*.

Предварительная ручная оценка результатов работы метода показывает, что метод дает хорошие результаты. Полученный автоматически список терминов достатчоно велик и должен рассматриватся как первый шаг к построению бОльших списков. С другой стороны, количество терминов в текстах ограничено, то есть такой список не может расти бесконечно.

В качестве будущих направлений работы можно упомянуть следующие:

- Определить возможность автоматического определения порога при отборе терминов.
- Попробовать разные параметры алгоритма классификации *k-means*. Оценить возможность применения алгоритма, позволяющего определять количество классов автоматически.
- Вместо меры сходства по косинусу угла попробовать другие меры сходства при классификации.
- Сравнить разные логарфимические меры сходства при поиске терминов.
- Выполнить сравнение с одним или несколькими корпусами разных предметных областей, чтобы отфильтровать общенаучные термины.

Заметим, что имплементация метода не представляет каких-либо существенных трудностей.

Литература

1. *Caraball S. A.* Automatic construction of a hypernym-labeled noun hierarchy from text. // In: Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics, 1999.
2. *Cimiano P.* Ontology learning and population from text, algorithms, evaluation and applications. // New York, USA: Springer, 2006.
3. *Dunning T.* Accurate methods for the statistics of surprise and coincidence. // Computational Linguistics 19.1 (Mar. 1993), 61–74.
4. *Gelbukh A., Sidorov G.* On Automatic Morphological Analysis of Inflective Languages. // In: Proc. of International Conference on computational linguistics and its applications Dialogue-2005 (in Russian), 2005, Russia, pp 92–96.
5. *Gómez-Pérez A., Fernandez-López M. & Corcho O.* Ontological Engineering. // London: Springer Verlag, 2004.
6. *Maedche A., Staab S.* Discovering conceptual relations from text. // In: Proceedings of ECAI 2000, 2000.
7. *Punuru J.* Knowledge-based methods for automatic extraction of domain-specific ontologies. // PhD thesis, 2007.
8. *Rayson P., Berridge D. and Francis B.* Extending the Cochran rule for the comparison of word frequencies between corpora. // In: Volume II of Purnelle G., Fairon C., Dister A. (eds.) Le poids des mots: Proceedings of the 7th International Conference on Statistical analysis of textual data (JADT 2004), Louvain-la-Neuve, Belgium, March 10–12, 2004, Presses universitaires de Louvain, pp. 926–936.
9. *Tingting He, Xiaopeng Zhang, Ye Xinghuo.* An Approach to Automatically Constructing Domain Ontology. // PACLIC 2006, Wuhan, China, 1–3 November, 2006, pp. 150–157.
10. *Uschold M. & Grunninger M.* Ontologies: Principles Methods and Applications. // Knowledge Egineering Review, 1996.