Извлечение семантических отношений из статей Википедии с помощью алгоритмов ближайших соседей

А. И. Панченко[2,1], С.А. Адейкин[1], А.В. Романов[1] и П.В. Романов[1]

{panchenko.alexander, adeykin90, jgc128ra, romanov4400}@gmail.com

[1] МГТУ им. Н.Э. Баумана, каф. Системы Обработки Информации и Управления

[2] Catholic University of Louvain, Center for Natural Language Processing (CENTAL)

Аннотация. В данной работе представлены методы извлечения семантических отношений из статей Википедии с помощью алгоритмов ближайших и взаимных ближайших соседей и двух метрик семантической близости. Мы производим анализ методов и приводим результаты их работы. Точность извлечения с помощью одного из методов достигает 83%. Кроме этого, мы представляем систему с открытым исходным кодом, которая эффективно реализует описанные алгоритмы.

Ключевые слова: семантические отношения, извлечение информации, Википедия, KNN, мера семантической близости

Введение

Существует множество типов *семантических отношений* между словами – синонимы, меронимы, антонимы, ассоциации и т.п. В рамках данной работы под семантическими отношениями понимаются синонимы, гиперонимы и ко-гиперонимы (слова имеющие общий гипероним). Подобные отношения успешно применяются в задачах *автоматической обработки текста*, таких как разрешение омонимии [1], расшире-

Игнатов Д.И., Яворский Р.Э. (ред.): Анализ Изображений Сетей и Текстов, Екатеринбург, 16-18 марта, 2012

ние поискового запроса [2], классификация текстовых документов [3] или создание вопросно-ответных систем [4]. Семантические отношения фиксируются в различного типа лингвистических ресурсах, к числу которых относятся, прежде всего, тезаурусы, онтологии, терминологические классификаторы и словари синонимов. Однако существующие ресурсы часто недоступны или недостаточны для конкретного приложения, предметной области или языка. При этом ручное создание требуемых семантических ресурсов – крайне дорогостоящий и трудоемкий процесс. В связи с этим, актуальной задачей является разработка методов автоматического извлечения семантических отношений.

Распространенный подход к извлечению отношений основан на лексико-синтаксических шаблонах, создаваемых вручную [5]. Недостатками данного подхода является сложность написания правил извлечения и применимость правил только для одного языка. Подходы, основанные на дистрибутивном анализе [6,7], не требуют ручной работы, но показывают невысокие результаты в задаче извлечения отношений [8]. В то же время недавно были предложены метрики семантической близости между словами на основе Википедии (www.wikipedia.org), показывающие отличные результаты [9,10,11]. Википедия привлекательна для анализа, так как она достаточно полно покрывает основные предметные области и языки, а также постоянно пополняется пользователями. Однако в предыдущих исследованиях мало внимания было уделено применению метрик, основанных на Википедии, для извлечения семантических отношений.

Данная работа восполняет этот пробел и фокусируется на применении подобных метрик для извлечения отношений. Цель предлагаемого в данной статье метода – найти для множества входных слов C (к примеру, терминов заданной предметной области) пары семантически связанных слов R. Рассматриваемые методы не возвращают тип найденной связи т. е. $R \subseteq C \times C$. Метод, предлагаемый в данной статье, характеризуется эффективностью, применимостью для языков доступных в Википедии и достаточной точностью. Новизна нашей работы по сравнению с существующими исследованиями и разработками заключается в следующем:

1) Предложены, реализованы и проанализированы новые *методы извлечения семантических отношений* из текстов статей Википедии, основанные на алгоритмах ближайших и взаимных ближайших соседей и двух метриках семантической близости слов (косинусе угла между векторами определений и количестве общих лемм в определениях).

2) Разработана программная система Serelex с открытым исходным кодом (лицензия LGPLv3), эффективно реализующая предложенные методы.

3 Извлечение семантических отношений из статей Википедии с помощью алгоритмов ближайших соседей

Методы извлечения семантических отношений

Данный раздел организован следующим образом. Сначала описываются данные, с которыми мы работаем, и способ их предварительной обработки. Затем обсуждаются алгоритмы извлечения семантических отношений и метрики семантической близости. В заключении описываются основные детали реализации системы Serelex.

Данные и их предварительная обработка

В качестве входных данных алгоритмы извлечения отношений получают множество определений D, для каждого из входных слов $c \in C$. Мы используем данные, доступные на DBPedia.org, для того чтобы построить множество определений английских слов (мы не включаем в этом множество словосочетания)[1]. Для каждого входного слова мы строим множество пар «слово;определение», где «слово» – это точное название статьи Википедии, а «определение» – текст первого параграфа этой статьи (аннотация к статье или т. н. extended abstract).

Аннотации к статьям были предварительно обработаны. Во-первых, из текста была удалена разметка и специальные символы. Во-вторых, был произведен морфологический анализ с помощью анализатора TreeTagger [12], в результате чего каждое слово было представлено в виде тройки «токен#ЧАСТЬ-РЕЧИ#лемма», к примеру «proved#VVN#prove». Приведем пример части определения термина «axiom», представленного в подобном формате:

axiom; in#IN#in traditional#JJ#traditional logic#NN#logic ,#,#, an#DT#an axiom#NN#axiom or#CC#or postulate#NN#postulate is#VBZ#be a#DT#a proposition#NN#proposition that#WDT#that is#VBZ#be not#RB#not proved#VVN#prove or#CC#or demonstrated#VVN#demonstrate but#CC#but considered#VVN#consider to#TO#to be#VB#be either#RB#either self-evident#JJ#self-evident ,#,#, or#CC#or subject#JJ#subject to#TO#to necessary#JJ#necessary decision#NN#decision .#SENT#.

Эксперименты, описанные в данной работе, были произведены на материале статьей, заглавие которых представляет собой одно слово без цифр и специальных символов. Данным критериям соответствовали 327167 статей Википедии. Для наших экспериментов мы подготовили два набора данных – малый (содержащий определения 775 слов (824Кб)) и большой (содержащий определения 327167 слов (237Мб)). Полученные “определения”, скрипт предварительной обработки статей и результаты извлечения доступны по адресу:

http://cental.fltr.ucl.ac.be/team/~panchenko/def/.

[1] Мы использовали файл с расширенными аннотациями (long abstracts): http://downloads.dbpedia.org/3.7/en/long_abstracts_en.nt.bz2

Сенлар [19] и другие исследователи методов извлечения отношений [7] отмечают, что подходы, основанные на синтаксическом анализе, зачастую достигают более высоких результатов, чем подходы, использующие только морфологический анализ. Тем не менее, в нашей работе мы сознательно не используем синтаксический анализ по двум причинам. Во-первых, в силу его высокой вычислительной сложности. Во-вторых, в силу того, что применение глубокого лингвистического анализа делает метод извлечения менее робастным. Предыдущие исследования показывают, что парсеры для различных языков обладают радикально отличным качеством. Кроме этого, стандартные парсеры делают много ошибок при анализе имен собственных и технических терминов – лексических единиц, представляющих наибольший интерес при извлечении семантических отношений.

Алгоритмы извлечения семантических отношений

Методы извлечения семантических отношений, рассматриваемые в данной статье, основаны на компонентном анализе [13,14], принцип которого заключается в том, что семантически близкие слова имеют подобные определения. Предложенные алгоритмы используют одну из двух метрик подобия определений – количество общих слов [15] или косинус угла между векторами определений [16]. В качестве входных данных алгоритмы извлечения семантических отношений принимают множество слов C, между которыми необходимо вычислить отношения и их определения D. Допустим, что на вход алгоритму поступило 5 слов, т.е. $C = \{alligator, animal, building, house, telephone\}$. Тогда задача алгоритма – распознать множество семантических отношений $R = \{\langle alligator, animal\rangle, \langle building, house\rangle\}$ из всех 10 возможных пар слов.

Первый алгоритм вычисляет семантические отношения с помощью метода ближайших соседей KNN, второй – с помощью метода взаимных ближайших соседей MKNN (Mutual KNN). Единственный метапараметр алгоритмов – количество ближайших соседей k. Псевдокод алгоритмов представлен на Рисунке 1.

Работа алгоритмов состоит в следующем. Сначала вычисляется мера семантической близости всех возможных пар определений (строка 6). На основе вычисленного значения заполняем массив наиболее близких слов `Rmatrix` для каждого определения (строки 1-12). При этом мы поддерживаем число элементов этого массива равным `k` (количеству ближайших соседей) – это позволяет сильно сократить потребление памяти без потери информации о связности слов. После заполнения массива наиболее близких слов для каждого определения все что остается сделать для получения результирующего набора отношений `R` в методе KNN – просто заполнить выходное множество, для метода

MKNN – дополнительно проверить для каждого определения входит ли оно в массив наиболее близких слов свой пары и если входит – добавить в результирующее множество (строки 13-21).

Сложность разработанных алгоритмов пропорциональна количеству поданных на вход слов $|C|$. Временная сложность равна $O(|C^2|)$, пространственная сложность также пропорциональна количеству ближайших соседей k и равна $O(k|C|)$.

```
R = ExtractRelations(C, D, k, isMKNN)
Input: C – слова, D – определения слов, k – количество ближайших соседей,
isMKNN – если true использовать алгоритм MKNN, иначе KNN
Output: R – множество семантических отношений <c_i,c_j> in C X C
1.  //Вычисление попарной близости между всеми словами C
2.  Rmatrix = void
3.  for i=0; i<count(C); i++ {
4.          for j=i; j<count(C); j++ {
5.                  // Вычисляем семантическую близость двух слов
6.                  s_ij = similarity(D(i), D(j))
7.                  // Сохраняем наиболее подобные слова
8.                  if( count(Rmatrix(C(i))) < k || s_ij > min(Rmatrix(C(i)) ){
9.                          Rmatrix(C(i)).addOrReplaceMin(C(j))
10.                 }
11.         }
12. }
13. // Вычисление семантических отношений
14. R = void
15. foreach c_i in Rmatrix {
16.         foreach c_j in Rmatrix(c_i) {
17.                 if(!isMKNN || Rmatrix(c_j) contains c_i){
18.                         R.add(<c_i, c_j>)
19.                 }
20.         }
21. }
22. return R
```

Рисунок 1. Псевдокод алгоритмов извлечения семантических отношений KNN и MKNN.

Метрики семантической близости слов

Функция `similarity` (строка 6) в алгоритмах KNN и MKNN вычисляет меру семантической близости между двумя словами на основе подобия их определений. Чем больше семантическая близость, тем более близок «смысл» слов. Мы используем две функции подобия определений. Первая метрика – количество общих лемм в определениях d_i, d_j слов c_i, c_j без учета совпадений стоп слов:

$$similarity(c_i, c_j) = \frac{2\left|(d_i \cap d_j) \,/\, stopwords\right|}{|d_i| + |d_j|}.$$

Здесь числитель равен количеству общих слов в двух определениях без учета стоп-слов; $|d_j|$ – количество слов в определении d_j; $stopwords$ – множество стоп-слов.

Вторая метрика – косинус угла между векторами определений $\boldsymbol{f}_i, \boldsymbol{f}_j$ представляющих слова c_i, c_j:

$$similarity(c_i, c_j) = \frac{\boldsymbol{f}_i \cdot \boldsymbol{f}_j}{\|\boldsymbol{f}_i\| \, \|\boldsymbol{f}_j\|} = \frac{\sum_{k=1,N} f_{ik} f_{jk}}{\sqrt{\sum_{k=1,N} f_{ik}^2} \sqrt{\sum_{k=1,N} f_{jk}^2}}.$$

Здесь f_{ik} – частота леммы c_k в определении d_i. Обе метрики подобия используют леммы (к примеру, animals#NNS#**animal**), не учитывают совпадения стоп-слов и учитывают совпадения лемм только со следующими частями речи: VV, VVN, VVP, JJ, NN, NNS, NP (существительные, глаголы и прилагательные).

Программный комплекс Serelex

Программное решение реализовано в виде консольного приложения на языке C++ и доступно для платформ Windows и Linux. Система состоит из классов определения, компонентного анализа, класса глобальных переменных, а также из нескольких вспомогательных классов и функций (см. Рисунок 2).

Основные функции программы заключаются в (1) загрузке файлов стоп-слов и слов, между которыми нужно найти отношения C; (2) загрузке с учетом стоп-слов и слов C файла дефиниций D; (3) вычислении семантической близости; (4) формировании списка наиболее близких слов R.

Для повышения быстродействия при загрузке дефиниций каждому слову сопоставляется уникальный числовой идентификатор и в дальнейшем вся работа по сравнению слов ведется с ним – это позволяет во много раз повысить быстродействие программы. В программном комплексе широко используется Стандартная библиотека шаблонов (STL) языка C++, что позволяет быстро, удобно и просто организовать хранение данных и работу с ними. Система Serelex имеет открытый исходный код, доступный на условиях лицензии LGPLv3 по адресу https://github.com/jgc128/Serelex.

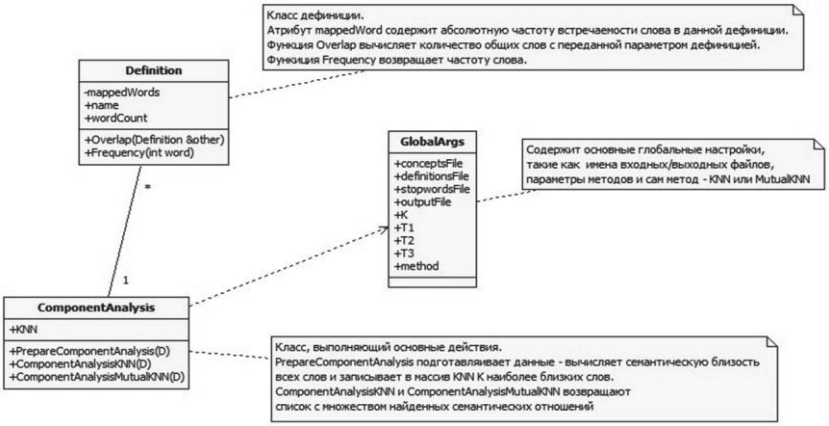

Рисунок 2. Основные классы системы извлечения семантических отношений Serelex.

Результаты

Мы исследовали работу алгоритмов KNN и MKNN с двумя описанными выше метриками близости и различными значениями количества ближайших соседей k (см. Рисунок 3). Полученные результаты свидетельствуют о практически линейном росте количества найденных отношений в зависимости от параметра k для обоих алгоритмов. При этом количество найденных отношений мало зависит от используемой метрики подобия. Алгоритм KNN извлекает больше отношений, чем MKNN, при равном количестве ближайших соседей k. Это происходит потому что MKNN удаляет пары, которые не являются взаимными соседями, в отличие от KNN.

Мы также провели оценку точности работы алгоритмов KNN и MKNN для $k = 2$ на множестве из 775 определений. Для этого мы разметили вручную файлы с извлеченными отношениям и вычислили точность извлечения как количество верных отношений к общему количеству извлеченных отношений. Результаты приведены в Таблице 1. Примеры извлеченных отношений между множеством из 775 слов с помощью алгоритма MKNN ($k = 2$) и количества общих слов в качестве метрики подобия приведены ниже[1]:

$R = \{\langle acacia, pine\rangle, \langle aircraft, rocket\rangle, \langle alcohol, carbohydrate\rangle,$
$\langle alligator, coconut\rangle, \langle altar, sacristy\rangle, \ldots \langle object, library\rangle,$

[1] Все извлеченные отношения с помощью данной конфигурации – http://cental.fltr.ucl.ac.be/team/~panchenko/def/results-775/overlap_mknn_2.csv

$\langle object, pattern\rangle, \langle office, crew\rangle, \langle onion, garlic\rangle, \langle saxophone, violin\rangle, \langle saxophone, clarinet\rangle\ \langle tongue, mouth\rangle, \ldots, \langle watercraft, boat\rangle, \langle watermelon, berry\rangle, \langle weapon, warship\rangle, \langle wolf, coyote\rangle, \langle wood, paper\rangle\}$.

В силу большого количества извлеченных отношений (см. Рисунок 3), оценка вручную качества извлечения для всех значений k затруднительна. Для больших значений k точность извлеченных отношений должна уменьшаться. При использовании метода мы рекомендуем использовать $k \in [1; 10]$. В будущем, мы планируем использовать WordNet [17] и стандартные проверочные наборы семантических отношений, такие как BLESS [18], для более точной оценки качества извлечения.

Скорость работы алгоритма при всех проведенных оптимизациях достаточно высокая. К примеру, 755 дефиниций обрабатываются чуть меньше чем за 3 секунды на сервере Linux 2.6.32-cs-kernel с процессорами типа Intel(R) Xeon(R) CPU E5606@2.13GHz (программа не использует многопоточность); алгоритм KNN при метрике подобия "количество общих слов" обрабатывает файл с 327167 дефинициями за 3 дня 3 часа и 47 минут.

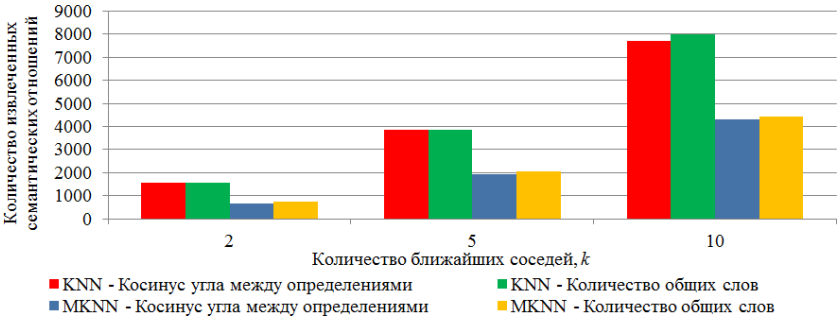

Рисунок 3. Зависимость количества извлеченных отношений от количества ближайших соседей k.

Таблица 1. Точность извлечения с помощью алгоритмов KNN и MKNN для $k = 2$ из 775 слов.

Алгоритм	**Мера подобия**	**Извлечено**	**Правильных**	**Точность**
KNN	Косинус угла	1548	1167	0.754
	Количество общих слов	1546	1176	0.761
MKNN	Косинус угла	652	499	0.763
	Количество общих слов	724	603	**0.833**

Обзор существующих методов

Сенлар [19] приводит обширный обзор методов извлечения семантических отношений основанных на дистрибутивном анализе и электронных словарях. Приводимые результаты оценивались на других лексиконах и поэтому не могут быть строго сравнены с нашими результатами, однако дают некоторую информацию об эффективности других систем и подходов. Так, система автоматического построения тезауруса SEXTANT извлекает отношения между существительными с точностью около 75%. Метрики семантической близости, основанные на Веб, достигают точности в задаче выбора синонима из четырех вариантов около 74%.

Метод и система, наиболее похожая на нашу работу, – это WikiRelate!, предложенная Струбе и Понзетто в 2006 году [9]. Основные отличия нашего подхода и системы заключаются в следующем:

- Serelex извлекает семантические отношения, а WikiRelate! вычисляет только меру связности слов;
- Serelex реализует две метрики близости (косинус и количество общих слов), а WikiRelate! только количество общих слов. При этом в совпадение n-грамма в данной системе считается как n^2 общих слов;
- Serelex не используют решетку категорий Википедии;
- Serelex может быть использован для вычисления отношений между определениями не только Википедии, но и других источников дефиниций, если они представлены в соответствующем формате;
- Исходный код системы WikiRelate! недоступен, а бинарная версия доступна только для использования в научных целях, в то время как Serelex имеет открытый исходный код и коммерчески дружественную лицензию LGPLv3.

В силу того что WikiRelate! не извлекает отношения, мы не можем напрямую сравнить ее эффективность с эффективностью Serelex. WikiRelate! достигает корреляции с суждениями человека о семантической близости до 0.59, однако корреляция равна 0.22, если система используется только текст статей без решетки категорий Википедии.

В работах [10] и [11] были представлены альтернативные метрики семантической близости между словами на основе текстов Википедии. Однако эти системы менее похожи на Serelex чем WikiRelate!. В частности, слова в них представляются в пространстве из всех статей Википедии, в то время как Serelex использует пространство лемм. Накаяма и др. [20] предложили еще один метод, основанный на Википедии, который значительно отличается от нашего – авторы используют структуру

гиперссылок между статей Википедии для извлечения отношений между словами. Наконец, Мильн и др. [21] предложили способы извлечения синонимов, гиперонимов и ассоциативных отношений из решетки категорий и других структурных и навигационных элементов Википедии.

Заключение

Мы предложили и исследовали методы извлечения семантических отношений из Википедии с помощью алгоритмов KNN и MKNN и двух метрик семантической близости. Предварительные эксперименты показали, что наилучшие результаты (83%) предоставляет метод, основанный на алгоритме MKNN и метрике подобия “количество общих слов”. Мы также представили систему с открытым исходным кодом, которая реализует описанные алгоритмы.

Предложенные методы характеризуются вычислительной эффективностью и достаточной точностью. Большой охват лексикона достигается за счет того, что слова представляются текстами статей Википедии. Поэтому метод потенциально применим для извлечения отношений между 3.8 миллионами терминов на английском языке и 17 миллионами терминов на других 282 языках Википедии (при наличии соответствующих морфологических анализаторов). Кроме этого, система Serelex может быть использована для извлечения отношений и между другими источниками определений, такими как Викисловарь или традиционные словари, если определения представлены в соответствующем входном формате.

Основные направления дальнейшего исследования следующие: (1) применение разработанного метода для извлечения отношений на русском, немецком и французском языках; (2) повышение точности извлечения за счет применения алгоритмов анализа структуры полученного графа семантических отношений между словами.

Благодарности

Работа была выполнена под руководством Юрия Николаевича Филипповича в рамках курса «Семиотика информационных технологий» МГТУ им. Н.Э. Баумана. Исследования Александра Панченко поддерживаются стипендией IN.WBI фонда Wallonie-Bruxelles International. Мы благодарим Ольгу Морозову, Ивана Зеленцова, Марину Даньшину, Екатерину Выломову и двух анонимных рецензентов за ценные комментарии.

Список источников

1 . Patwardhan S., Pedersen T. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. EACL , page 1-12, 2006

2 . Hsu M.H., Tsai M.F., Chen H.H. Query expansion with conceptnet and wordnet: An intrinsic comparison. Information Retrieval Technology, pages 1–13, 2006

3 . Tikk D., Yang J.D., Bang S.L. Hierarchical text categorization using fuzzy relational thesaurus. KYBERNETIKA-PRAHA, 39(5): 583–600, 2003.

4 . Sun R., Jiang J., Fan Y., Hang T., Tatseng K., Yen Kan C.M. Using syntactic and semantic relation analysis in question answering. In Proceedings of the TREC, 2005

5 . Hearst M.A., Automatic acquisition of hyponyms from large text corpora, Proceedings of the 14th conference on Computational linguistics COLING '92, 1992

6 . Lin D. Automatic retrieval and clustering of similar words. Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics, 768-774, 1998

7 . Heylen K., Peirsman Y., Geeraerts D., Speelman D. Modelling word similarity: an evaluation of automatic synonymy extraction algorithms. Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 3243-3249, 2008

8 . Curran J.R. and Moens M. Improvements in automatic thesaurus extraction. Proceedings of the ACL-02 workshop on Unsupervised lexical acquisition. 59-66, 2002.

9 . Strube, M. and Ponzetto, S.P., WikiRelate! Computing semantic relatedness using Wikipedia. Proceedings of the National Conference on Artificial Intelligence, 1419-1429, 2006.

10 . Gabrilovich E., Markovitch S. Computing semantic relatedness using wikipedia-based explicit semantic analysis. International Joint Conference on Artificial Intelligence, 12-20.2007.

11 . Zesch T., Müller C., Gurevych I. Extracting lexical semantic knowledge from wikipedia and wiktionary. In Proceedings of the LREC, pages 1646–1652, 2008.

12 . Helmut Schmid. Probabilistic Part-of-Speech Tagging Using Decision Trees. pages 44–49, 1994.

13 . Филиппович Ю.Н., Прохоров А.В., Семантика информационных технологий: опыты словарно-тезаурусного описания. Серия «Компьютерная лингвистика». М.:МГУП, 2002 http://it-claim.ru/Library/Books/CL/CLbook.htm

14 . Кобозева И. М. Компонентный анализ лексического значения. Лингвистическая семантика: 4-е изд. М.: Книжный дом «ЛИБРОКОМ», стр. 109-122, 2009.

15 . Banerjee S., Pedersen T. Extended Gloss Overlaps as a Measure of Semantic Relatedness,In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence, 2003.

16 . Jurafsky D., Manning H. M., An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Second Edition. 697-701, 2009.

17 . Fellbaum, C. WordNet. Theory and Applications of Ontology: Computer Applications, 231--243, Springer, 2010.

18 . Baroni, M. and Lenci, A. How we BLESSed distributional semantic evaluation. In Proceedings of GEMS 2011, 2011.

19 . Senellart P., Blondel V. D. Automatic Discovery of SimilarWords. Survey of Text Mining II.2008, 1, 25-44, Springer London, 2008.

20 . Nakayama K., Hara T., and Nishio S. Wikipedia Mining for an Association Web Thesaurus Construction, Web Information Systems Engineering – WISE, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 322-334, 2007.

21 . Milne D., Medelyan O., and Witten, I.H. Mining Domain-Specific Thesauri from Wikipedia: A Case Study. Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 442-448, IEEE Computer Society, 2006