Сравнительное тестирование авторубрикаторов

Ю. В. Титов, В. В. Фарсобина

В данной работе описаны методика проведения сравнения и полученные результаты для трех авторубрикаторов текстов, в основу разработки которых были положены три различных алгоритма классификации. Целью данного исследования не было глубокое и полное исследование рубрикаторов, однако все же хотелось получить данные, достаточные для принятия решения о целесообразности развития и/или поддержки одного или нескольких из них.

Введение

Авторубрикатор — это программа, которая работает в двух режимах — в режиме обучения и в режиме рубрикации (классификации). Термин «рубрикация» мы здесь и далее будем употреблять в том же смысле, что и «классификация», подразумевая лишь, что происходит классификация текстовых документов по нескольким рубрикам.

В режиме обучения на вход программе подаются текстовые (или в формате HTML) документы, про которые сказано, к какой рубрике они относятся. Набор рубрик при этом тоже заранее задан. В этот момент происходит обучение — построение некоторой модели (набора данных), которые в дальнейшем помогут произвести рубрикацию.

В режиме же классификации на вход подаются документы, рубрика которых заранее неизвестна, и программа должна определить наиболее релевантную рубрику. Релевантность — это характеризующая данный документ численная величина, показывающая, насколько данный документ подходит к той или иной рубрике.

Рубрикатор считается тем лучше, чем меньше ошибок он допускает при классификации [7]. Также немаловажным является возможность рубрикатора уметь определять те документы, которые не подходят ни под одну рубрику.

Описание алгоритмов, лежащих в основе тестируемых авторубрикаторов

Перечислим алгоритмы, на основе которых разрабатывались рубрикаторы.

1) базовый алгоритм;
2) алгоритм «Наивный Байес»;
3) алгоритм SVM — Метод Опорных Векторов.

Базовый алгоритм был разработан несколько лет назад [4, 6], успешно использовался в задачах классификации текстов и взят нами для сравнения с другими алгоритмами в качестве отправной точки — чтобы было с чем сравнивать другие алгоритмы. Релевантность документов в нем измеряется целым числом от 0 до 100. Если алгоритм считает, что документ не относится ни к какой рубрике, то возвращается 0.

Однако практика показывает, что документы, имеющие значения, близкие к нулю, как правило, тоже далеки от данной рубрики. В работающей системе уровень, ниже которого документы считаются не подходящими к этой рубрике, принят равным 15. Это пороговое значение мы далее будем называть Null_Level (нулевой уровень).

Второй авторубрикатор был реализован на основе широко известного алгоритма «**Наивный Байес**» [1, 2]. Например, этот алгоритм используется в одном из модулей фильтрации спама популярной почтовой программы The Bat! [8].

«Наивной» же она называется, так как исходит из предположения о взаимной независимости признаков, и как ни странно, этого часто оказывается вполне достаточно.

Релевантность возвращаемая этим рубрикатором такая же, как и у базового алгоритма — целое число от 0 до 100. Однако, значения релевантности нельзя сравнивать без определенного пересчета.

Для Наивного Байеса Null_Level мы взяли за 45. Это значение было подобрано опытным путем: при дальнейшем его увеличении точность алгоритма начинала быстро падать, в то время как при уменьшении Null_Level ниже 45 алгоритм начинал совсем плохо определять «чужие» документы, т. е. не принадлежащих ни одной рубрике.

Третий рубрикатор — на основе **алгоритма SVM** («Support Vector Machine» или «Метод Опорных Векторов»). Этот алгоритм был предложен Владимиром Вапником еще в 70-х гг., и в дальнейшем значительно им

же развивался [5]. В настоящий момент различные модификации и доработки SVM получили весьма широкое распространение.

Для построения классификатора нами была использована программа SVM-light [3]. Особенностью этого алгоритма является его бинарность, т. е. при классификации он отделяет только два класса друг от друга, и для построения мультиклассифкации приходится производить дополнительные усилия. Возвращаемая релевантность — действительное число в окрестности нуля. Для этого рубрикатора основное значение Null_Level мы взяли равным 0.

В данной реализации был применен метод отделения одной рубрики от всех. Тем самым, для классификации по 10 рубрикам происходило одновременное обучение сразу 10 рубрикаторов. Результатом классификации была рубрика с наибольшей релевантностью.

Если же релевантность документа для всех рубрик оказалась меньше Null_Level, то считалось, что документ «чужой», т. е. не принадлежит ни какой рубрике.

Параметры авторубрикаторов для оценки

Как уже говорилось, целью данного исследования не было полное и досконально изучение характера поведения рубрикаторов во всех ситуациях. Опишем сначала исследуемые параметры. В заключении статьи упомянем также то, что тоже было бы интересно и полезно посмотреть, однако выпало из данной работы.

Были получены и оценены следующие характеристики:

1) точность и полнота обнаружения «чужих» документов;
2) точность и полнота классификации;
3) F-мера для пунктов 1) и 2).

Сформулируем общие определения:

Точность — это отношение правильно полученных документов ко всем полученным.

Полнота — отношение правильно полученных документов ко всем правильным.

F-мера — это среднее пропорциональное полноты и точности

$$= 2 / (1/\text{Полнота} + 1/\text{Точность}).$$

В нашем исследовании мы считали, что документ найден, если ему присвоена некоторая рубрика. В противном случае говорилось, что документ не найден, и он «чужой». Исключением было тестирование на нахождение «чужих» документов. В этом случае все было наоборот — найденными считались документы, которым не было присвоено ни одной рубрики.

Так же исследовалось поведение рубрикаторов в зависимости от количества документов в обучающей выборке и от количества различных рубрик.

Тестовая выборка

В качестве основного рабочего материала использовалась тестовая выборка, состоящая из 10 различных рубрик:

- Авто;
- Ближний_восток;
- В_мире;
- В_России;
- Кавказ;
- Культура;
- Медицина;
- Спорт;
- Террор;
- Экономика.

Список рубрик и тексты документов брались из новостей сайта Lenta.ru в период с января по апрель 2005 г.

В каждой рубрике было примерно одинаковое количество документов — от 250 до 300, что обеспечивало равномерность абсолютных результатов — никакая из рубрик не выделялась только из-за количества документов в ней. Хотя, конечно, точность нахождения документов для отдельной рубрики может сильно зависеть от специфики документов именно этой рубрики. Например, в данном наборе рубрика Авто очень хорошо отделялась от других практически всеми классификаторами, а вот В_России, Кавказ и Террор сильно путались.

Общее количество документов в выборке составило 2 668 документов. Они были разделены случайным образом на 2 равные части по 1 334

документа в каждой, с сохранением примерно равного количества документов по рубрикам. Итого получилось примерно по 135 документов в каждой из десяти рубрик.

Обучение проводилось на одном из этих двух наборов, а тестирование — на другом. После этого наборы менялись местами и прогон повторяли. Во всех же итоговых данных (графиках, таблицах) фигурировало среднее арифметическое этих прогонов. Данное усреднение позволяет сгладить результаты — тем самым сделать их более объективными.

Методика проведения тестирования

Для проверки возможности нахождения авторубрикаторами «чужих» документов нами был применен следующий прием: из обучающей выборки удалялись документы какой-то одной рубрики, они в обучении не участвовали. Однако, при тестировании документы этой рубрики присутствовали.

Если документ из выкинутой рубрики определялся как «чужой», то считалось, что это правильно найденный «чужой».

Было проведено 10 тестов — при отсутствии поочередно каждой из рубрик. На итоговой диаграмме по горизонтальной оси отмечены как раз удаленные рубрики.

При такой схеме для каждого документа возможно 5 различных исходов:

1) «Отл»: документ из какой-то рубрики («Свой») правильно определился в свою рубрику;
2) «Чуж»: действительно «Чужой» документ определился как «Чужой»;
3) «Ошиб»: документ из какой-то рубрики определился не в свою рубрику;
4) «Св_чуж»: «Свой» документ ошибочно определился как «Чужой»;
5) «Чуж_св»: «Чужой» документ ошибочно определился как «Свой» — т. е. попал в какую-то рубрику.

Первые два пункта — правильная работа авторубрикатора, а последние три — ошибочные исходы.

Тем самым, если произвести суммирование по всем документам тестовой выборки, то для нахождения «Чужих» документов получаем:

$$\text{Точность} = \text{Чуж} / (\text{Чуж} + \text{Св\_чуж}).$$

(Пояснение: мы ищем «чужие» документы.)

$$\text{Полнота} = \text{Чуж} / (\text{Чуж} + \text{Чуж\_св})$$

Для оценки же классификации документов между рубриками получаем:

$$\text{Точность} = \text{Отл} / (\text{Отл} + \text{Ошиб} + \text{Чуж\_св})$$

$$\text{Полнота} = \text{Отл} / (\text{Отл} + \text{Ошиб} + \text{Св\_чуж})$$

Напомним, что F-мера — это среднее пропорциональное полноты и точности

$$= 2 / (1/\text{Полнота} + 1/\text{Точность}).$$

Основные результаты

В сводку основных результатов включены данные рубрикации для трех алгоритмов, однако, для SVM добавлен еще один прогон. Дело в том, что параметр Null_Level позволяет регулировать соотношение Точность/Полнота. А именно: за счет уменьшения точности, можно увеличивать полноту и наоборот. Это свойственно всем описанным авторубрикаторам, в той или иной степени.

Для SVM на графиках представлены данные для значения Null_Level 0,0 и 0,1. Во втором случае мы жертвуем точностью, увеличивая полноту.

Ниже, на графиках представлены результаты десяти тестов. В каждом тесте в обучающей выборке отсутствовала одна из рубрик. На всех диаграммах по горизонтальной оси как раз и отмечены те рубрики, которые отсутствовали в обучающей выборке. По вертикальной оси отмечены значения соответствующих характеристик.

Вначале точность и полнота при нахождении «чужих» документов (рис. 1, 2), затем точность и полнота при нахождении «своих» (рис. 3, 4).

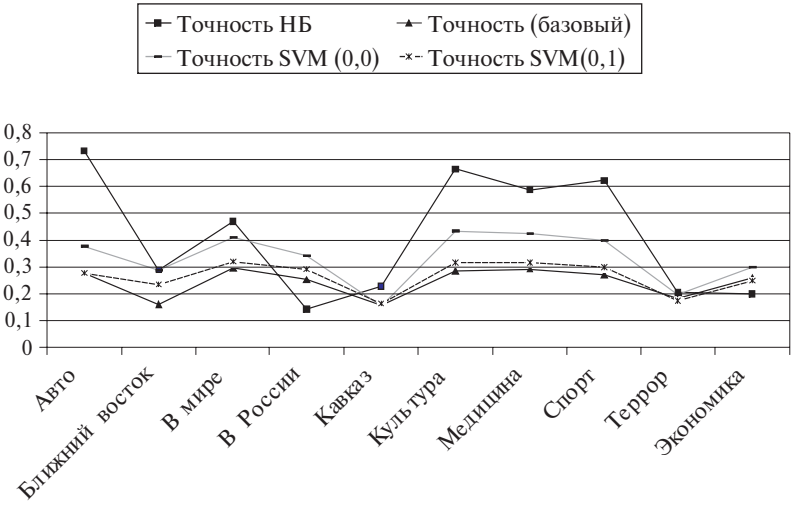

Рис. 1. Точность нахождения «чужих» документов

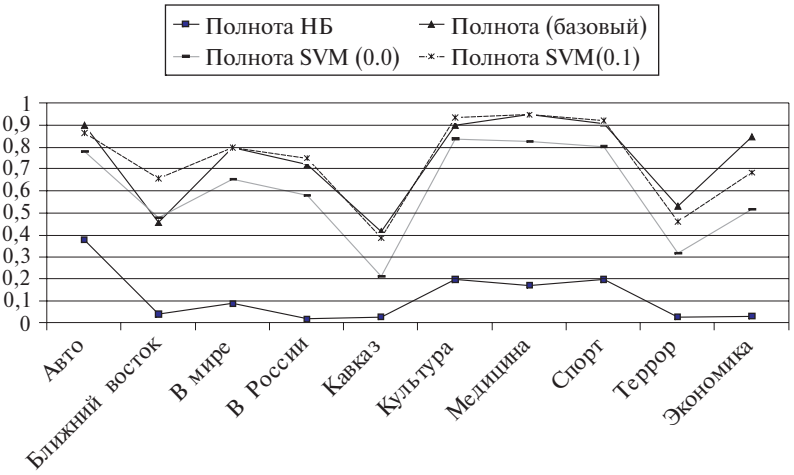

Рис. 2. Полнота нахождения «чужих» документов

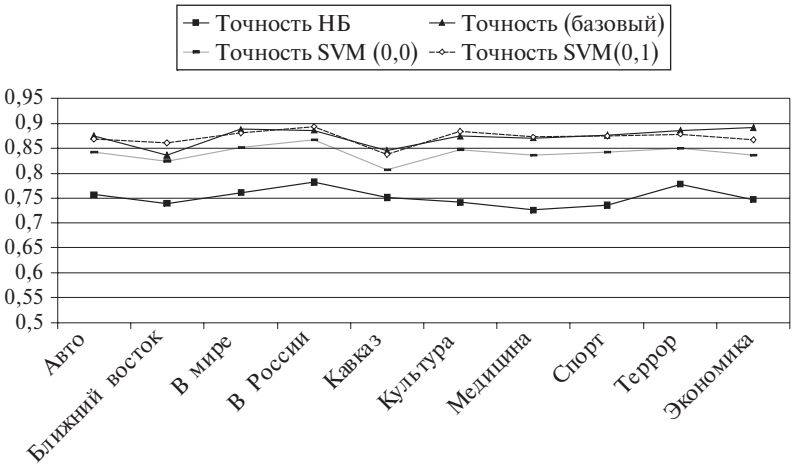

Рис. 3. Точность нахождения и классификации «своих» документов

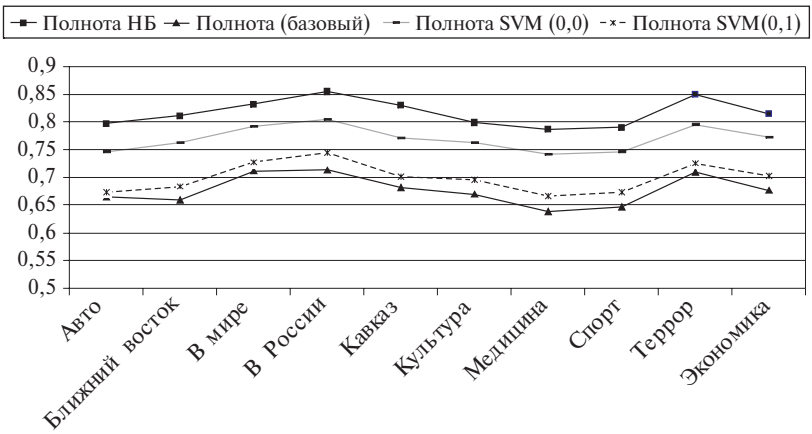

Рис. 4. Полнота нахождения и классификации «своих» документов

Наконец, F-мера — единая метрика для характеристики точности и полноты (рис. 5, 6).

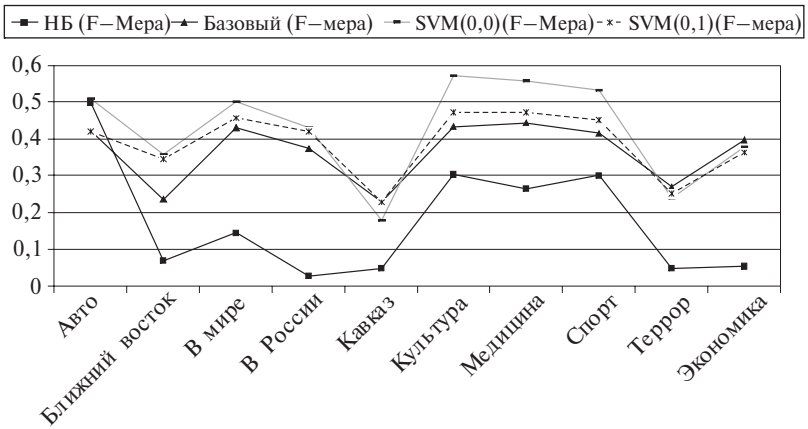

Рис. 5. F-мера при нахождении «чужих» документов

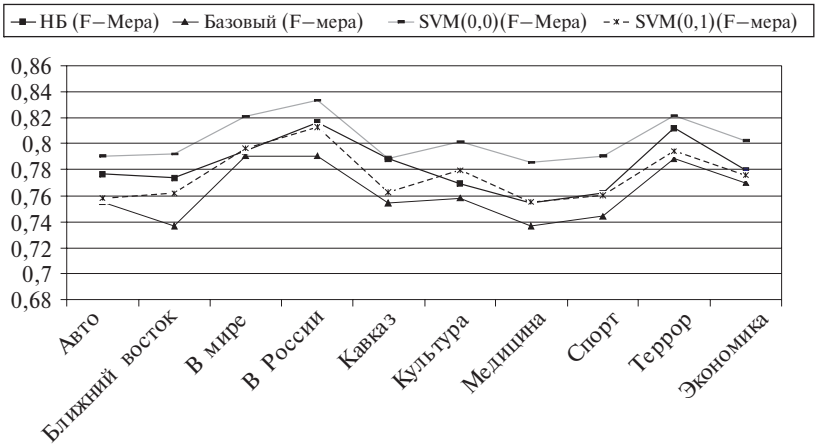

Рис. 6. F-мера нахождения и классификации «своих» документов

Зависимость от количества документов

Было проведено исследование зависимости точности рубрикаторов от размера обучающей выборки. Этот тест проводился из предположения,

что «чужих» документов нет, т. е. у каждого документа есть определенная рубрика — достигался такой эффект путем установки Null_Level в минимальное значение.

Тестовая выборка при этом использовалась другая, отличная, от описанной выше. По составу же рубрик они практически идентичны.

На графике представлена зависимость количества ошибок (в процентах от общего числа документов в тестовой выборке) от количества документов в обучающей выборке (рис. 7, 8).

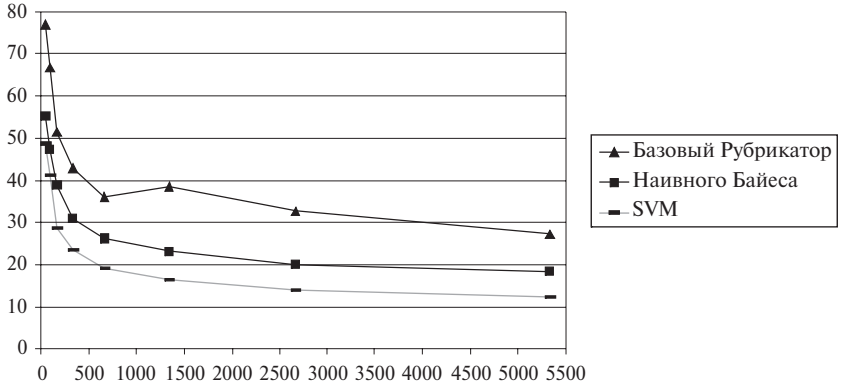

Рис. 7. Количество ошибок в процентах

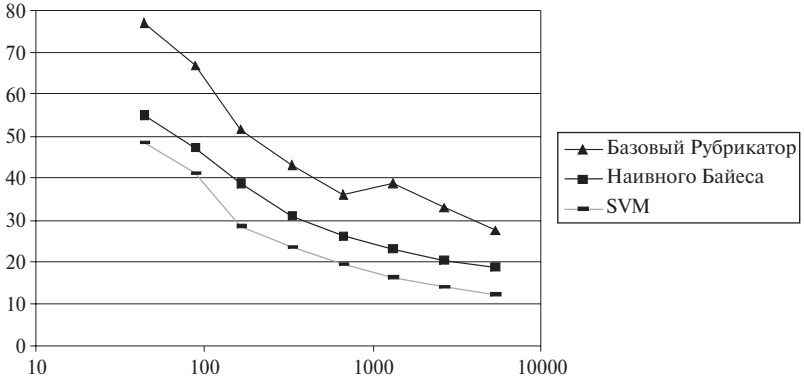

Рис. 8. Тот же самый график,
только горизонтальная шкала — логарифмическая

Количество документов изменялось от 44 (по 4 документа в каждой рубрике — при обучении) до 5 337. Данные усреднялись — бралось среднее арифметическое среди нескольких тестов, причем, чем меньше документов в обучающей выборке, тем большее количество тестов для нее производилось.

Так, для обучающего набора из 168 документов было сделано 32 прогона. При этом в каждом прогоне обучающая выборка была различна, а вот тестовый набор совпадал.

Зависимость от количества рубрик

Так же была проведена серия тестов, целью которой ставилось узнать — насколько падает точность классификации при увеличении количества рубрик. Приводим сравнительные результаты тестов под номерами 1, 2, 3 и 4 в выборках которых присутствовало соответственно 2, 4, 6 и 8 различных рубрик. Вот эти рубрики, в том порядке, в котором они добавлялись в выборку: Авто и Ближний_восток, В_мире и Культура, В_России и Медицина, Спорт и Экономика.

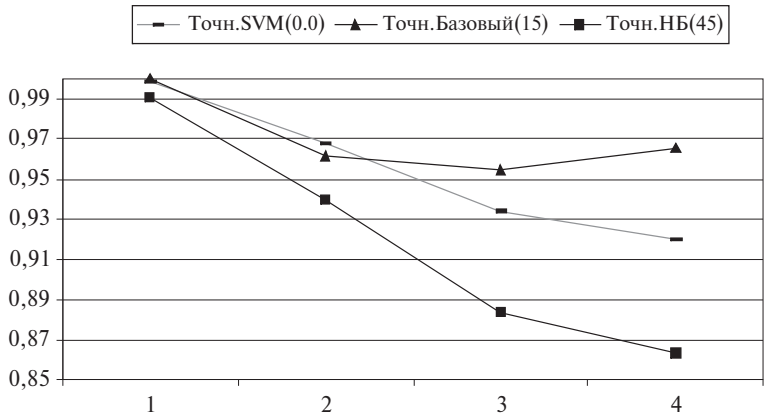

Рис. 9. Точность

В скобках легенды указано значение Null_Level (0, 15 и 45) (рис. 9, 10, 11).

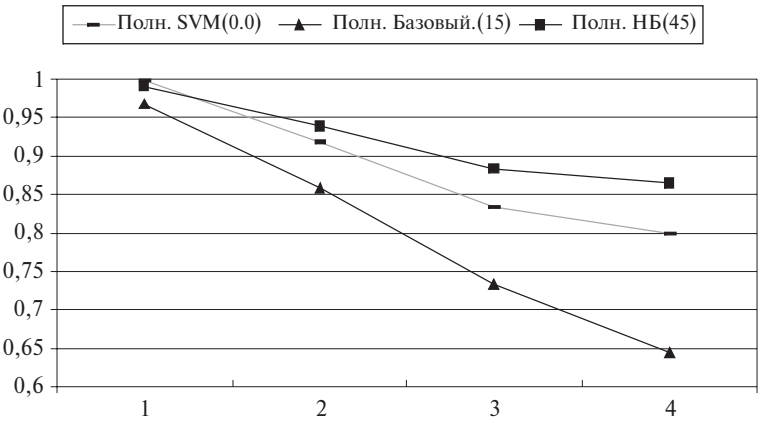

Рис. 10. Полнота

Видно, что у всех рубрикаторов точность и полнота с ростом количества рубрик уменьшаются. У базового алгоритма точность несколько возрастает, однако поводов для радости нет — так как полнота у него при этом резко уменьшается.

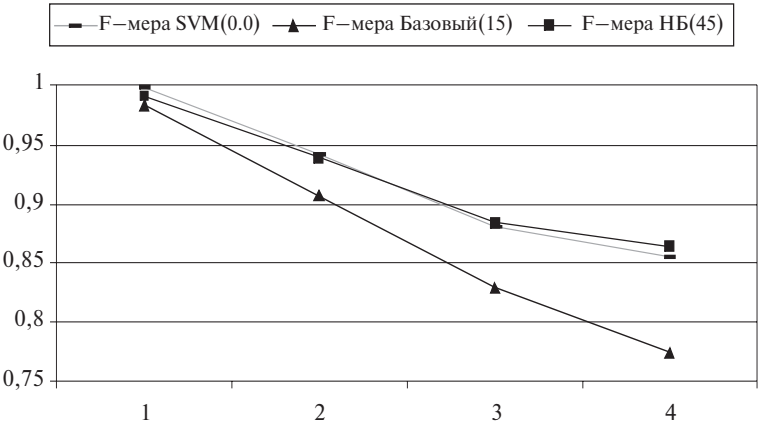

Рис. 11. F-мера

В случае же когда Null_Level стоит в минимуме, получается следующая картина (рис. 12).

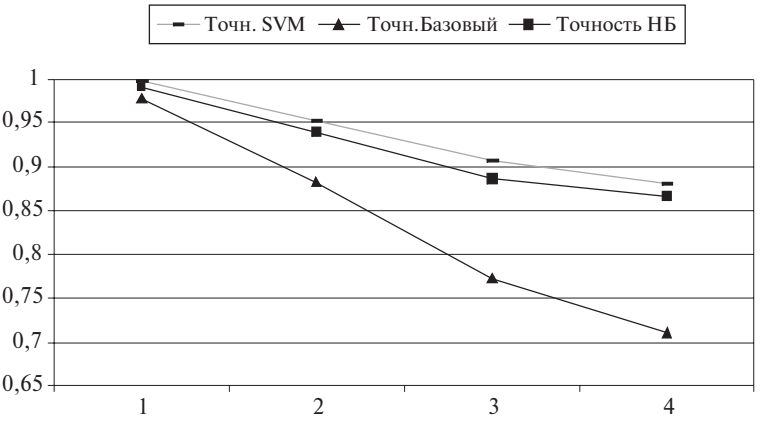

Рис. 12

Выводы

После анализа имеющихся данных, можно сделать следующие выводы. Рубрикатор на основе базового алгоритма достаточно неплох — в ситуации, когда надо отсеивать «чужие» документы. Именно это и требуется на практике в подавляющем большинстве случаев. В этой ситуации ему заметно проигрывает алгоритм на основе Наивного Байеса. Однако, в ситуации, когда заранее известно, что у каждого документа есть рубрика, базовый начинает сильно проигрывать двум другим алгоритмам.

SVM в целом показывает себя не хуже как базового, так и «Наивного Байеса». В тестах «Свой — Чужой» SVM совсем немного превосходит базовый, и сильно отрывается вперед по сравнению с Байесом.

Стоит сделать рад оговорок. Во-первых, тестирование проводилось фактически на одном наборе рубрик. Поведение рубрикаторов при существенно отличающемся наборе может быть иным. Например, там, где сейчас один рубрикатор сильно превосходит другие — эта разница может свестись к минимуму, и наоборот. Хотя кардинальное изменение всей картины маловероятно.

Во-вторых, в отличие от базового рубрикатора, который достаточно долго отлаживался, на разработку двух других было затрачено относительно небольшое время. При некоторых доработках эти рубрикаторы могут начать превосходить базовый алгоритм в большей степени.

Заключение

Приведем возможные направления для исследования и дальнейшего сравнительного тестирования авторубрикаторов.

1. Иерархия — как себя ведут рубрикаторы при иерархической рубрикации?
2. Рубрикация документов сразу в несколько рубрик — на практике часто один и тот же документ относится сразу к несколькими рубрикам. Как повлияет нахождение одного и того же документа одновременно в обучающих выборках нескольких рубрик?
3. Рубрикация при исчезающе малом количестве «Своих» документов — часто бывает необходимо «найти иголку в стоге сена». Кто в этом случае лучше?

Литература

1. Classification using Naive Bayes
   // http://www.resample.com/xlminer/help/NaiveBC/classiNB_intro.htm
2. Naive Bayes algorithm for learning to classify text.
   // http://www-2.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html
3. *Thorsten J.* SVM-light Support Vector Machine
   // http://www.cs.cornell.edu/People/tj/svm_light/
4. *Дягилева А. В., Киселев С. Л., Сомин Н. В.* Статистическая модель рубрикации текстов на примере сообщений СМИ // Дистанционное образование. 1998. № 7. С. 16–21.
5. *Мерков А. Б.* Основные методы, применяемые для распознавания рукописного текста, глава 4.
   // http://www.recognition.mccme.ru/pub/RecognitionLab.html/methods.html
6. *Сомин Н. В., Соловьева Н. С., Соловьев С. В.* Система рубрикации текстовых сообщений // Труды Междунар. Семинара Диалог'98 по компьютерной лингвистике и ее приложениям / Под ред. А. С. Нариньяни. Казань: Хэтер, 1998. В 2 т. Т. 2. С. 574–581.
7. Труды РОМИП'2004, Санкт-Петербург: НИИ Химии СПбГУ, 214 с. Сентябрь 2004 / Под ред. И. С. Некрестьянова.
   http://romip.narod.ru/romip2004/index.html
8. Фильтрация спама по Байесу //
   http://www.ritlabs.com/ru/solutions/bayesyan.php