УДК 004.912

Д.В.Дивенкова, Л.Е.Мигунова, Н.В.Филимоненкова
Санкт-Петербургский политехнический университет Петра Великого

ИСПОЛЬЗОВАНИЕ МЕТРИК ПРИ РАСПОЗНАВАНИИ СЛОВОФОРМ

Область исследования. Работа посвящена проблеме автоматического (машинного) распознавания словоформ. Словоформа – это слово в определенной грамматической форме, которое образовано путем спряжения или склонения главной словарной формы слова. К примеру, существительное в русском языке может иметь до двенадцати словоформ (шесть падежей в единственном и множественном числе): «порог», «порога», «порогу», «пороги», «порогами» и т.д..

Актуальность исследования. Выделение пары слов, которые являются словоформами одного и того же слова, актуально для следующих IT-задач: исправление ошибок и опечаток в текстовых данных, автозамена при наборе текста, автоматический перевод, перепись населения, индексация текста и др.

Цель и методы исследования. Словоформы одного и того же слова, как правило, фонетически схожи (отличаются только окончанием, суффиксом или приставкой). Поэтому для распознавания словоформ можно использовать строковые метрики. Цель данного проекта – оценить эффективность метрик Хэмминга, Левенштейна, триграмм и Джаро–Винклера в задаче распознавания словоформ. Методы исследования – численный эксперимент и статистический анализ.

Классические определения этих метрик описаны во втором столбце табл. 1. При этом функции Хэмминга и Левенштейна являются метриками в строго математическом смысле, т.е. удовлетворяют аксиомам метрики, тогда как функции триграмм и Джаро–Винклера можно назвать метриками только условно. Недостатком метрики Хэмминга является то, что она задается на множестве слов одинаковой длины, что сильно ограничивает область ее применения. Поэтому при вычислении расстояния Хэмминга мы добавляли пробелы в конце более короткого слова, чтобы оба слова стали одинаковой длины.

Мы провели нормировку данных метрик по методу, предложенному в работе [1] для метрики Левенштейна. В результате построены функции H', L', ${G_3}'$, J'(табл. 1, третий столбец), которые удовлетворяют аксиомам метрики и принимают значения от 0 до 1: 0 – слова совпали, 1 – слова не имеют ничего общего с точки зрения данной метрики.

Таблица 1. Определения метрик

Название метрики	**Классическое описание метрики** (x, y – строки)	**Нормированная версия метрики** ($\|x\|$, $\|y\|$ – длины строк)
метрика Хэмминга	$H(x,y)$ – число позиций, в которых соответствующие символы строк x и y различны. Пример: H (дворник_, дворянин)=4	$H'(x,y) = \frac{H(x,y)}{\max\{\|x\|,\|y\|\}}$ Пример: H'(дворник_, дворянин)=0,5
метрика Левенштейна	$L(x,y)$ – минимальное число операций «вставка», «замена», «удаление» символа, необходимых для преобразования строки x в y. Пример: L (дворник, дворянин)=2	$L'(x,y) = \frac{L(x,y)}{\max\{\|x\|,\|y\|\}}$ Пример: L'(дворник, дворянин)=0,25

метрика триграмм	$G_3(x,y)$ – число общих триграмм (трех, идущих подряд, символов) в строках x и y. Пример: G_3 (дворник, дворянин)=2	$G_3{}'(x,y) = 1 - \frac{2G_3(x,y)}{G_3(x,x) + G_3(y,y)}$ Пример: $G_3{}'$ (дворник, дворянин)=0,64
метрика Джаро–Винклера	$J(x,y)$ – мера схожести слов, путем нахождения одинаковых символов в двух словах, которые находятся на расстоянии, составляющим не более половины длины самого длинного слова, с учетом перестановок совпадающих символов. J(дворник, дворянин)=1,13	$J'(x,y) = 1 - J(x,y)$ Пример: J'(дворник, дворянин)=0,13

Результат. На языке C# в среде разработки MS Visual Studio составлена программа для проведения численного эксперимента и анализа его результатов. Опишем сначала алгоритм программы в целом.

Входными данными программы является каталог слов X: 8 базовых слов (в главной словарной форме), к каждому из них подобрано 5 словоформ – всего 48 слов в каталоге. Программа рассматривает все пары слов (x,y) из X, т.е. всего 2016 пар. Среди них 240 пар состоят из слов x и y, которые являются словоформами одного из базовых слов. Например, «порога» и «порогами» – это две словоформы слова «порог». Назовем это множество пар «классом словоформ» и обозначим X_0, тогда все остальные пары принадлежат «классу несловоформ» – X_1.

Для каждой пары слов (x,y) из X программа вычисляет значения метрик H', L', $G_3{}'$, J'. Если значение меньше заданного порога t, то программа относит пару (x,y) к классу X_0, в противном случае – к классу X_1. Порог t можно задать в диапазоне от 0 до 1. Ясно, что чем ниже порог t, тем меньше слов программа «назовет» словоформами.

Поскольку мы знаем, какие пары слов на самом деле относятся к X_0, то все 2016 результатов делятся на четыре группы – в зависимости от сочетания реального класса и класса, предписанного метрикой (табл. 2).

Таблица 2. Матрица ошибок классификации

Реальный класс	**Класс, предписанный метрикой**	
	Negative	**Positive**
Negative	TN (True Negative) – количество пар слов, которые метрика отнесла к X_1, и они действительно из X_1.	FP (False Positive) – количество пар слов, которые метрика отнесла к X_0, но на самом деле они из X_1.
Positive	FN (False Negative) – количество пар слов, которые метрика отнесла к X_1, но на самом деле они из X_0.	TP (True Positive) – количество пар слов, которые метрика отнесла к X_0, и они действительно из X_0.

Для анализа эффективности метрик в программе используются стандартные показатели качества бинарного классификатора, [2] – [4]. В первую очередь это *Precision* (точность) и *Recall* (полнота) – числовые значения, которые вычисляются для данного каталога X, данной метрики и данного порога t.

Precision (точность) – вероятность того, что слова действительно словоформы при условии, что метрика определила их как словоформы. Иными словами, это способность метрики отличать данный класс от других классов:

$$Precision = \frac{TP}{TP + FP}$$

Recall (полнота) – вероятность того, что метрика определит слова как словоформы при условии, что они действительно словоформы, т.е. способность метрики обнаруживать данный класс:

$$Recall = \frac{TP}{TP + FN}$$

И *Precision*, и *Recall* принимают значения от 0 до 1: чем ближе к 1, тем лучше метрика справляется со своей задачей.

Существует несколько способов синтезировать данные *Precision* и *Recall*. В частности, построить график PR-кривой, отражающий зависимость *Precision* от *Recall* через параметр *t:*

$$PR - \text{кривая: } \begin{cases} x = Recall(t) \\ y = Precision(t) \end{cases}, \text{где } t - \text{порог.}$$

Площадь под графиком PR-кривой показывает эффективность метрики в целом, т.е. при любых пороговых значениях t [5]. Та метрика считается качественней, чья площадь под графиком больше. У идеальной метрики площадь под графиком PR-кривой равная 1. Именно этот инструмент анализа позволил нам получить наиболее выразительные результаты.

Описанную программу мы протестировали на четырех каталогах *X*:

- каталог существительных,
- каталог прилагательных,
- каталог глаголов,
- каталог фонетически похожих существительных.

Ограничение каждого каталога одной частью речи было предпринято для того, чтобы выявить связь между эффективностью метрики и грамматическими правилами образования словоформ, которые для каждой части речи свои. Каталог из фонетически похожих существительных (например, «лабиринт» – «лаборант») и их словоформ был протестирован, чтобы усложнить метрикам задачу. Во всех случаях мы рассматривали словоформы без приставки, чтобы метрика Хэмминга оставалась конкурентноспособной.

На рисунках 1–4 приведены графики PR-кривых для метрик H', L', ${G_3}'$, J' (соответствующие кривые отличаются цветом) и каждого из четырех каталогов.

Значения площадей под графиком PR-кривой для метрик в каждом каталоге словоформ представлены в табл. 3.

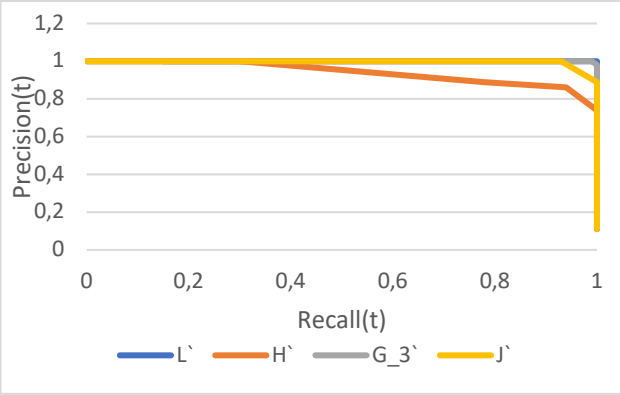

Рисунок 1. Графики PR-кривых для существительных

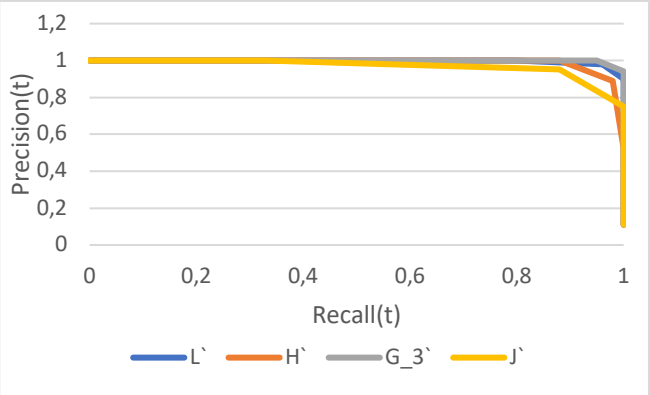

Рисунок 2. Графики PR-кривых для прилагательных

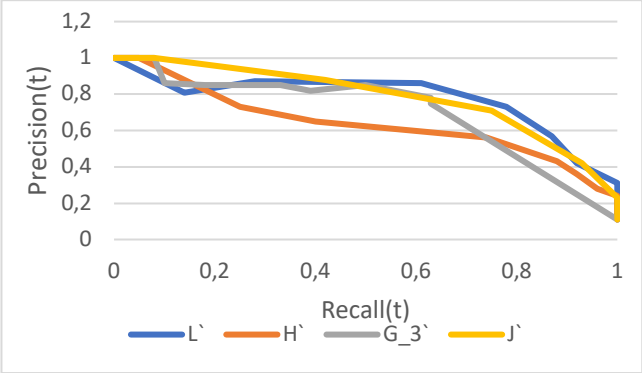

Рисунок 3. Графики PR-кривых для глаголов

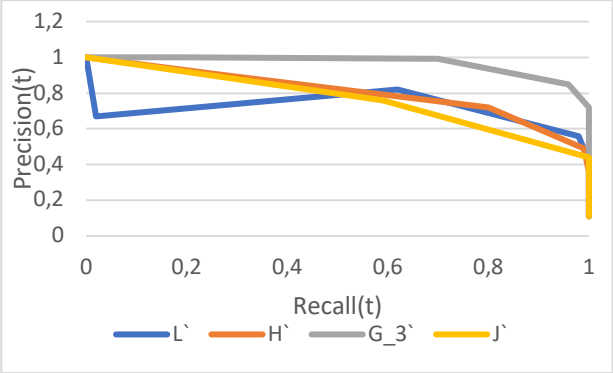

Рисунок 4. Графики PR-кривых для похожих существительных

Таблица 3. Значения площади под графиками PR-кривых

	Существит.	Прилагательные	Глаголы	Похожие сущ.
метрика Хэмминга H'	0,6837	0,988	0,64	0,806
метрика Левенштейна L'	1	0,996	0,78	0,722
метрика триграмм ${G_3}'$	0,999	0,998	0,7015	0,968
метрика Джаро–Винклера J'	0,996	0,968	0,79	0,765

Сначала сопоставим результаты для каталогов существительных, прилагательных и глаголов, в которых базовые слова брались без фонетического сходства. Все метрики однозначно лучше справляются с обнаружением словоформ среди существительных и прилагательных, нежели среди глаголов. Причиной является особенность русского языка: словоформы глаголов могут очень сильно отличаться, например, «спать» – «сплю».

При этом для каталога существительных лучший результат показала метрика Левенштейна (причем оказалась идеальной), для каталога прилагательных – метрика триграмм, для каталога глаголов – метрика Джаро–Винклера.

Теперь сравним результаты для каталога фонетически похожих существительных и каталога существительных без фонетического сходства: в первом случае PR-кривые всех метрик проходят ниже и площади под графиком меньше, чем во втором. Это обусловлено тем, что в первом случае метрикам, естественно, сложнее отличить словоформы от несловоформ. Метрика триграмм лучше всего справилась с этой усложненной задачей.

Чтобы выяснить среди метрик абсолютного лидера, мы рассчитали для каждой метрики среднее арифметическое площадей под графиками PR-кривых по всем каталогам (табл. 4).

Таблица 4. Среднее арифметическое площадей под графиками PR-кривых

	метрика Хэмминга	метрика Левенштейна	метрика триграмм	метрика Джаро–Винклера
S	0,779	0,874	0,917	0,880

Выводы. Численный эксперимент показал, что метрика триграмм является наиболее эффективной в задаче распознавания словоформ, метрика Хэмминга – наименее эффективной.

ЛИТЕРАТУРА

1. Лиманова Н. И., Седов М. Н. Алгоритм идентификации реквизитов физических лиц в базах данных на основе метрики Левенштейна// Научно-технический выпуск информационных технологий, механики и оптики. 2012. №6(32). С. 136 – 140
2. Davis J., Goadrich M.The Relationship Between Precision-Recall and ROC Curves// University of Wisconsin-Madison
3. Maria del Pilar Angeles, Adrian Espino-Gamez. Comparison of methods Hamming Distance, Jaro and Monge-Elkan// Universidad Nacional Autonoma de Mexico Mexico, D.F// IARIA. 2015.
4. Почему DataScientist-ы не используют ошибки первого и второго рода. URL: https://habrahabr.ru/post/340048/ (дата обращения: 01.12.2018)
5. Соколов Е. Семинары по выбору моделей. 15.04.2015. – С. 5–8