ИЛЬЯ СЕГАЛОВИЧ И РАЗВИТИЕ ИДЕЙ КОМПЬЮТЕРНОЙ ЛИНГВИСТИКИ В ЯНДЕКСЕ

Зеленков Ю. Г. (yuryz@yandex-team.ru),
Зобнин А. И. (alzobnin@yandex-team.ru),
Маслов М. Ю. (maslov@yandex-team.ru),
Титов В. А. (uht@yandex-team.ru)

Яндекс, Москва, Россия

В статье рассматриваются наиболее важные и интересные лингвистические проекты, в которых участвовал и которыми руководил Илья Сегалович (1964–2013), один из создателей поисковой системы Яндекс. Среди этих проектов: разработка морфологического анализа и синтеза русских слов, позволяющего обрабатывать «новые» слова, не включенные в словарь; снятие морфологической омонимии для русского языка с помощью нормализующих подстановок; практическая транскрипция иностранной собственной и нарицательной лексики; автоматическая расстановка ударений и анализ поэтических текстов; создание эффективных методов распознавания нечетких дубликатов для текстовых документов; разработка информационно-справочной системы «Национальный корпус русского языка» и др. Описываются ключевые идеи и подходы, связанные с поиском решений сложных лингвистических задач и рассказывается о роли Ильи в изобретении этих подходов и их дальнейшем развитии. Приводятся примеры нетривиальных лингвистических алгоритмов, созданных Ильей вместе с коллегами.

Ключевые слова: Илья Сегалович, Яндекс, морфология, практическая транскрипция, анализ стихов, нечеткие дубликаты

ILYA SEGALOVICH AND DEVELOPMENT OF IDEAS OF COMPUTATIONAL LINGUISTICS TO YANDEX

Zelenkov Yu. G. (yuryz@yandex-team.ru),
Zobnin A. I. (alzobnin@yandex-team.ru),
Maslov M. Yu. (maslov@yandex-team.ru),
Titov V. A. (uht@yandex-team.ru)

Yandex, Moscow, Russia

In the article the most important and interesting linguistic projects led by Ilya Segalovich (1964–2013) — one of the founders of the Yandex search engine — are considered. He also took part in their development. The following projects are among them. Development of the morphological analysis and synthesis of Russian words with a possibility of processing «new» words not included in the dictionary; solving the problem of morphological ambiguity for the Russian language with the help of normalizing substitutions; practical transcription of foreign, individual and common words; automatic positioning of stresses and the analysis of poetic texts; creation of efficient methods of recognizing fuzzy duplicates for textual documents; development of the information and require system «The National Corpus of Russian», etc. Key ideas and approaches connected with the searching of solutions to complicated linguistic problems are described, and Ilya's role in the invention of these approaches and their further development is stated. Examples of non-trivial linguistic algorithms developed by Ilya in collaboration with his colleagues are given.

Key words: Ilya Segalovich, Yandex, morphology, practical transcription, analysis of poems, fuzzy duplicates

Введение

Илья Сегалович (1964–2013) внес большой вклад практически во все основные поисковые технологии Яндекса. И все-таки именно компьютерная лингвистика всегда была для него предметом особого интереса, как исследовательского, так и практического.

Основной чертой Ильи, как IT-профессионала, была уникальная способность находить и четко формулировать наиболее важные задачи, которые необходимо решать в данный момент, и в области информационного поиска в целом, и в компьютерной лингвистике в частности. Предлагаемые им алгоритмы всегда отличались оригинальным подходом и основывались на энциклопедическом знании той предметной области, к которой они относились. Уровень образованности Ильи поражал. Он всегда был в курсе самых последних публикаций практически по любому вопросу. Причем это касалось не только материалов текущих конференций, но и книжных новинок (в первую очередь зарубежных).

По инициативе и под руководством Ильи был выполнен перевод на русский язык одной из самых популярных в мире книг «Введение в информационный поиск» Кристофера Д. Маннинга и др. [7]. Специально для этой книги Ильей, совместно с известными специалистами, была проведена большая работа по созданию, упорядочиванию и редактированию современного толкового терминологического словаря для предметной области информационного поиска. В России работа такого объема была выполнена впервые.

Он придумал и был на протяжении ряда лет основным организатором конкурсов «Интернет-математика» и «Класс», которые существенно стимулировали проведение у нас в стране научно-исследовательских и прикладных разработок, а также создание учебных курсов в области компьютерного анализа данных. Под его редакцией вышли также несколько сборников этих конкурсов [5, 6].

Илья был одним из вдохновителей и первых участников проекта «Национальный корпус русского языка» [9, 12]. Благодаря именно его инициативе компания Яндекс в начале 2000-х вложилась в этот проект как финансово, так и людьми, осуществив поддержку разработки силами Виталия Титова, Андрея Кондратьева, Андрея Аброскина и других и предоставив поисковый модуль Яндекс.Сервер. Система снятия омонимии [3] и новая система морфологического разбора Яндекса, в которой усовершенствована обработка несловарных слов, обучены на корпусе со снятой омонимией из НКРЯ.

Илья Сегалович является автором алгоритма открытого (т. е. позволяющего с высокой точностью обрабатывать не входящие в словарь слова) морфологического анализа и синтеза для нескольких языков — ключевого лингвистического инструмента поисковой технологии Яндекса [10].

Помимо успешного решения этой базовой задачи компьютерной лингвистики Илья принимал активное участие в целом ряде других проектов, как чисто лингвистических, так и общепоискового характера, но опирающихся на лингвистику. О некоторых наиболее важных и интересных из таких проектов и пойдет речь дальше.

1. Морфологический анализ и синтез

В 1995–1996 гг. разработчики подразделения «Аркадия» фирмы Comptek International под руководством Ильи делали программную оболочку для электронных научных изданий «Грибоедов» и «Информ-Норматив». Главной задачей было создание полнотекстовой поисковой системы, и мы участвовали в формировании поискового конкорданса, т. е. в данном случае списка нормальных форм всех слов, встречающихся в текстах корпусов.

При построении конкорданса обнаружилось, что для «Информ-Норматива» «стандартный» морфологический словарь Зализняка покрывает список словоформ корпуса всего лишь наполовину. Это было довольно неожиданно для нас. Ведь корпус текстов относительно невелик, грамматически корректен, поэтому мы не ожидали в нем большого лексического разнообразия. Стала понятной важность задачи морфологической обработки слов, не содержащихся в словаре.

В тот период Илья предложил алгоритм морфологического анализа и синтеза для таких слов [10]. Новизна его идеи в сравнении с [2] состояла в «синтетической» части алгоритма, т. е. в возможности генерации гипотез моделей словоизменения для новых слов. Гипотезы строились в формате системы ЭТАП [8].

Проблема в том, что обычно таких гипотез получалось больше одной и нужен был способ выбора лучшей. Мы попробовали использовать информацию из корпуса текстов «Информ-Норматива», сравнивая парадигмы гипотез. Т.е. для гипотезы строили список словоформ, встречающихся в корпусе текстов, и применяли две несложные эвристики — «включение парадигмы» и «наличие нормальной формы» [10]. Поскольку корпус текстов «Информ-Норматива» был небольшим (да и эвристики немудреными), они срабатывали всего примерно в 20 % случаев. Так что этот результат мог быть только небольшим подспорьем специалистам, окончательно формировавшим конкорданс.

Программа MyStem, первая версия которой была написана Ильей Сегаловичем и Виталием Титовым, умеет строить такие гипотетические разборы для слов, не входящих в словарь.

Возможность получить еще более хороший результат для обработки как словарных, так и «несловарных» слов — с приписыванием значимых весов гипотезам разбора –возникла позже, с появлением и достаточным наполнением корпуса со снятой омонимией НКРЯ. Мы планируем в ближайшее время выпустить очередную версию программы — MyStem 3.0, поддерживающую ранжирующую морфологию и контекстное снятие омонимии.

Процедура ранжирования разборов на основе корпуса НКРЯ является дальнейшим развитием идей, предложенных Ильей в более ранних версиях морфологического алгоритма.

Для осуществления такого ранжирования применяется машинное обучение по корпусу. Просто упорядочить список разборов — плохой вариант, так как корпус является достаточно разреженным, и не каждое слово в нем представлено. Поэтому мы вычисляем по корпусу следующие величины: частоту каждой морфологической схемы (парадигмы), а внутри схемы — частоты каждой основы и каждого окончания слова. Каждому разбору конкретного слова соответствует своя схема, точно определяющая границы основы и окончания. Мы предполагаем, что события «встретилась данная основа *stem*» и «встретилось данное окончание *flex*» слова *word* при фиксированной схеме разбора *scheme* независимы. Поэтому вероятность разбора слова *word* по схеме *scheme* можно определить по формуле Байеса:

$$P(scheme \mid word) = \frac{P(word \mid scheme)P(scheme)}{P(word)} = \frac{P(stem \mid scheme)P(flex \mid scheme)P(scheme)}{P(word)}.$$

Этот подход представляет собой наивный байесовский классификатор. Его преимущество состоит, во-первых, в том, чтоданные о частотах распределены по основам и окончаниям (и поэтому их можно эффективно упаковать), а во-вторых, в том, что разреженность корпуса и низкие частоты отдельных форм слов уже не представляют проблемы: эти данные будут сглажены за счет других форм с такой же основой и других слов с таким же окончанием внутри данной схемы. Кроме того, полученные частоты дополнительно сглаживаются (например, простейшим методом Лапласа).

Точность этой модели на русском языке достигает 95,9% (по тексту леммы), в то время как точность простого выбора леммы с самой частотной схемой (baseline) равна 90%. Обращаем внимание на то, что здесь не используется контекст слова.

2. Проблема снятия омонимии при автоматической обработке текстов

Снятие омонимии полезно во многих приложениях компьютерной лингвистики, в частности, в поисковых системах оно может повысить точность

обработки запросов и сократить объем хранимой в архивах информации. К сожалению, эффективных подходов к решению этой задачи для русского языка на тот момент (2003 г.) не существовало (да и сейчас положение не намного лучше), и поэтому Илья предложил разработать новый оригинальный алгоритм решения этой задачи, опирающийся на информационные возможности Яндекса [3].

Для целей поиска необходимо было в первую очередь научиться снимать неполную (или морфологическую) омонимию, при которой разные слова совпадают не во всех, а только в нескольких грамматических формах, т. е. являются омоформами (имеют разные леммы). Примеры: три, стекло, стих, стали, белка, пара, вина, кос и др. Тем не менее полученные результаты позволяли (с небольшими изменениями) применять данный алгоритм и к разрешению полной (или лексической) омонимии.

Было принято решение использовать машинное обучение с учителем, поскольку основной целью являлась максимизация точности получаемых результатов. В дальнейшем появилась идея расширить алгоритм возможностью обучения без учителя, так как это существенно облегчало его настройку и обобщение подхода на другие языки.

В качестве данных для обучения алгоритма сначала использовался аннотированный массив текстов с морфологической разметкой и снятой омонимией из проекта «Национальный корпус русского языка» [9]. Затем для повышения полноты обучающей выборки был создан дополнительный веб-корпус. Процедура отбора текстов для этого корпуса учитывала поисковые возможности Яндекса и была автоматизирована с помощью специально разработанного генетического алгоритма, который обеспечивал оптимальную репрезентативность выборки документов из веба сразу по нескольким наиболее важным показателям (максимальное разнообразие явлений омонимии, жанров публикаций и их тематик).

При построении корпуса использовалась идея ранжирования самых частотных омонимов русского языка по степени «трудности выбора леммы». Омоним считался более «трудным», если для его разрешения с точностью выше фиксированного порога 0,96–0,97 требовался больший набор обучающих примеров для включения в корпус. Процедура ранжирования была полностью автоматизирована, и это позволило существенно (в разы) минимизировать размер обучающей выборки, не снижая качества получаемых результатов. В результате было выделено и ранжировано около 25 тыс. различных омонимов вместе с их контекстами.

Далее было нужно решить важную задачу удобного представления контекстов — слов, окружающих омоним слева и справа в предложении. Илья предложил оригинальную идею (полностью оправдавшую себя в дальнейшем) записывать отдельные элементы контекста в виде нормализующих подстановок, указывающих, сколько букв нужно отнять у словоформы с конца и на что их заменить, чтобы получить лемму. У флективных языков (к которым принадлежит и русский) нормализующие подстановки являются достаточно универсальным средством выражения грамматических свойств, поэтому мы и решили использовать именно их в качестве элементов контекста омоформы.

Затем возникла проблема количественной оценки «силы влияния» элемента контекста на выбор нужной леммы в зависимости от того, слева или справа от омонима расположен данный элемент и на каком расстоянии (сосед, через слово, ...). Кроме того, нужно было решить, сколько таких элементов брать, чтобы и словарь был достаточно компактным, и точность не пострадала. Была придумана функция, которая позволяла упорядочить элементы контекста по степени их влияния на выбор леммы. Оказалось, что наибольшим влиянием обладает сосед слева, затем сосед справа, затем следующие два элемента слева и следующий сосед справа. Сила влияния остальных слов резко падала, и поэтому они отбрасывались, т. е. при построении словаря учитывались только пять окружающих слов в указанном порядке, выраженном весовыми коэффициентами. На основе описанных процедур был построен словарь контекстов объемом около 150 тыс. обучающих примеров.

Алгоритм прошел своеобразную закалку во время одного из конкурсов «Интернет-математика», когда ему пришлось участвовать в поединке с алгоритмом, предложенным одним из участников. На основе проведенных тестов и экспертных оценок алгоритм Яндекса одержал победу.

Точность алгоритма составляет порядка 0,96–0,97 и при необходимости легко может быть увеличена с помощью автоматизированного дообучения. В настоящее время алгоритм входит в состав леммера — основного лингвистического инструмента Яндекса, работающего с 17 языками и у истоков которого также стоял Илья.

3. Практическая транскрипция собственной и нарицательной лексики

Современный интернет, основанный на идеологии Веб 2.0, характеризуется большим числом новостных потоков, множеством социальных сетей, развитием блогосферы, цифровой картографии, увеличением доли аудио- и видеоконтента и т. п. Все это приводит к существенному возрастанию удельного веса имен собственных как в веб-документах, так и в поисковых запросах. Как правило, эти имена написаны на самых различных языках и обозначают личные имена людей, географические названия, бренды, музыкальные группы и их произведения, фильмы и т. п.

Успешная обработка имен собственных для целей поиска осложняется тем, что в их написании царит произвол: имена, обозначающие одни и те же сущности, часто имеют несколько вариантов написания, как русскими буквами, так и латинскими.

Для наведения порядка в этом хаосе Илья в 2008 г. предложил разработать универсальный алгоритм практической транскрипции (записи русскими буквами максимально близкого звучания иностранного имени) собственной лексики. Задача была довольно амбициозная. Во-первых, правила транскрипции должны были быть вероятностными и составляться автоматически, т. к. ручные подходы в принципе не могли охватить всего разнообразия языковых

явлений. Статистический характер правил позволял создавать многовариантные гипотезы и тем самым мог как-то справиться с проблемой неоднозначности. Во-вторых, алгоритм должен был уметь определять исходный язык имени (или несколько языков, упорядоченных по вероятности) независимо от того, кириллицей или латиницей записано имя. В-третьих, нужно было научиться делать транскрипцию и в обратную сторону — с русского на язык оригинала. И наконец, необходимо было (хотя бы частично) научиться транскрибировать и нарицательную лексику, поскольку многословные имена собственные довольно часто ее включают (Empire State Building).

Для придания спортивного драйва проекту Илья даже пригласил внешнего разработчика со своими конкурентными идеями, но тот не выдержал предложенного темпа и постепенно сошел с дистанции.

За основу алгоритма было взято машинное обучение с помощью транскрипционных билингв (и в отдельных случаях, мультилингв) — параллельной записи имени на двух и более языках. Необходимое число билингв (порядка 100 тыс.) было получено автоматически из архивов Яндекса и открытых веб-источников, прежде всего из Википедии. Билингвы были написаны на 17 языках и включали все основные типы имен (имена людей, топонимы, бренды, ...)

Было перепробовано много вариантов алгоритма транскрипции, но наиболее удачной (простой в реализации, независимой от языка и эффективной по результатам) оказалась идея «метода сегментов», которую Илья и выбрал для последующей реализации и которая удовлетворяла всем вышеперечисленным требованиям к алгоритму. Сегмент — это группа рядом стоящих гласных или согласных букв, выделение которой для любого языка (за исключением *тайского*) тривиально. Правила транскрипции отдельного сегмента можно также легко получить из обучающей выборки следующим образом.

Сначала выполняем побуквенное выравнивание билингвы: «s t a — t e m e n t» = «с т е й т — м е н т», затем собираем сегменты и убираем пропуски (если они есть) в левой части: «st a t e m e nt» = «ст ей т — м е нт», и наконец убираем пропуски в правой части, объединяя сегменты в левой: «st a tem e nt» = «ст ей тм е нт». На практике все происходит гораздо проще, т. к. в подавляющем большинстве случаев (до 95 %) сегменты уже находятся во взаимно-однозначном соответствии и выравниваются без проблем.

Далее полученные по всей выборке соответствия сегментов группируются вместе с соседними сегментами слева и справа (учет контекста повышает точность), подсчитываются их частоты и окончательно формулируются вероятностные правила «перевода». В процессе транскрипции исходное слово разбивается на сегменты с контекстами, для каждого сегмента выбираются все варианты «перевода», вероятности суммируются, и результаты ранжируются по убыванию сумм вероятностей. Подход полностью симметричен как в направлении от латиницы к кириллице, так и обратно.

Алгоритм практической транскрипции используется в общепоисковых задачах и музыкальном поиске. В 2013 г. он был с успехом применен в сервисе «Яндекс.Карты» для перевода мировой карты примерно с 40 языков (включая тайский с его весьма специфической графикой для записи гласных) на русский.

«Метод сегментов» может быть успешно применен и для распознавания языков. Так, в 2010 г. было проведено соревнование по определению языка документа между существовавшим в Яндексе алгоритмом и новым, созданным на основе сегментов и дополненным механизмом языковых сигнатур, позволяющим легко разделять такие языки, как, например, русский и болгарский или шведский и датский. По тестам для 31 языка со счетом 28:3 победил новый алгоритм.

4. Автоматическая расстановка ударений и определение размера стиха

Интерес к автоматическому анализу поэтических текстов возник у Ильи примерно в 2009 г., когда в Яндексе был запущен проект «Стихолюб», а НКРЯ к тому времени содержал уже достаточно представительный размеченный поэтический подкорпус. Но это все были, так сказать, пассивные методы работы со стихами, требующими участия людей, а Илью в первую очередь привлекала возможность компьютерного подхода к проблеме.

Так возникла идея создать автономную (независимую от других лингвистических инструментов типа морфологии) и не очень сложную процедуру, которая с хорошей точностью могла бы распознавать ритмические фрагменты в текстах, определять размер стиха, рифмовку и пр.

Первая проблема на этом пути — автоматическая расстановка ударений в любых (в том числе отсутствующих в словаре) словах, т. к. от ее успешного решения зависели все остальные этапы. Была использована идея корреляции между буквенными концами слов и позицией ударения. С помощью машинного обучения был построен словарь буквенных концов объемом около 300 тыс. единиц. Вероятность правильной расстановки ударения с помощью такого словаря составляла почти 0,99 для нарицательной лексики и немного меньше для имен собственных.

Затем была написана довольно точная процедура фонетической транскрипции, и можно было приступать к автоматическому анализу стихов. Для классического силлабо-тонического стихосложения и некоторых неурегулированных размеров (дольник, тактовик, …) проблема, как ни странно, оказалась не очень сложной. Были записаны структурные формулы для этих размеров и для анализируемого стиха (после расстановки ударений) проверялась «близость» к этим формулам с помощью особой метрики. Самая близкая формула и определяла искомый размер и клаузулу (без ошибок анализировалось, например, стихотворение В. Брюсова «Ночь»). Результаты были довольно точные, программа «понимала», что такое спондей, пиррихий и другие премудрости типа синкопы икта, правда похуже.

Для улучшения результатов дополнительно использовался анализ строфы как целого. Илья придумал специальную функцию, которая могла ранжировать различные варианты определения структуры строфы по их внутренней «симметрии» и тем самым повышать качество разбора отдельных стихов, входящих в строфу. Например, в отдельных случаях нужно было смещать стандартное ударение («и г`орьки мне, горьк`и твои упреки»). С помощью специальной

таблицы также достаточно надежно определялась схема рифмовки (с попытками, где нужно, замены «е» на «ё»).

На этапе тестирования с помощью созданной процедуры было обнаружено несколько десятков тысяч ошибок в поэтической разметке НКРЯ (в основном расстановка ударений и определение размера).

Илье очень понравились полученные результаты, и он хотел внедрить анализатор стихов в разные сервисы типа «Яндекс.Словари», но тогда не сложилось. Совсем недавно на основе этой процедуры был создан «Автопоэт» — программа, которая сочиняет «стихи» произвольной формы (онегинская строфа, шекспировский сонет, ...) из запросов пользователей [1] (а сам Илья еще в 2011 г. предлагал сочинять стихи из твитов).

5. Обнаружение нечетких дубликатов в веб-документах

Распознавание нечетких дубликатов актуально для большого количества современных веб-приложений: это и улучшение качества индекса и архивов поисковиков за счет удаления избыточной информации, и объединение новостных сообщений в сюжеты на основе их сходства по содержанию, и фильтрация спама (как почтового, так и поискового), и установление нарушений авторских прав при незаконном копировании информации, и ряд других.

К 2007 г. уже было разработано множество интересных алгоритмов решения этой задачи [4], включая алгоритм, в создании которого участвовал и сам Илья [14]. Но все эти алгоритмы, без исключения, имели плохие показатели полноты, примерно 0,50–0,60, хотя точность была высокой — порядка 0,90–0,95. Поэтому Илья и предложил заняться повышением полноты без потери точности.

Самый надежный путь для этого — попарное сравнение (например, с помощью расстояния Левенштейна) всех документов из архива поисковика — был невозможен чисто физически. В Яндексе уже тогда было более 1 млрд документов, и число сравнений получалось астрономическое, измеряемое квинтиллионами. Нужно было что-то придумать. И тогда родилась идея провести декомпозицию этой гигантской матрицы, разложив ее на множество мелких подматриц, внутри которых выполнение попарных сравнений уже некритично.

Алгоритм декомпозиции оказался довольно простым. Нужно было для каждых пяти соседних слов (т. н. «шинглов») каждого документа указать длину документа в словах. Затем объединить одинаковые шинглы в последовательности по возрастанию длин документов. Было установлено, что, если соседние длины отличаются больше, чем на 15 %, эти документы не могут быть дубликатами. Таким образом, с помощью этого простого соотношения цепочки документов распадались на небольшие группы, внутри которых прямым попарным сравнением с помощью редакционного расстояния находились полудубликаты. Эксперимент показал 99%-ную точность и практически 100%-ную полноту при вполне приемлемой производительности.

Для дополнительного уменьшения размеров групп и повышения производительности, помимо числа слов в документах, использовалось также число

предложений, вместо «шинглов» брались 3 самых длинных предложения, а вместо расстояния Левенштейна сравнивались 5 самых длинных слов. При этом скорость существенно возрастала, а точность и полнота практически не менялись.

Результаты экспериментов: на коллекции РОМИП — полнота = 96%, точность=95%, F1-мера=0,95; на почтовой коллекции — полнота = 99%, точность=95%, F1-мера=0,97.

6. Оценка качества архивов поисковиков и определение объема Рунета

Несмотря на то что этот проект в меньшей степени связан с компьютерной лингвистикой, чем предыдущие, рассказать о нем необходимо, поскольку здесь ярко проявилась креативность Ильи, его способность к нестандартным подходам. В 2003–2004 гг. начался быстрый рост Рунета, и одновременно обострилась конкуренция между ведущими поисковиками. Для мониторинга этих процессов были необходимы автоматические процедуры, которые на регулярной основе могли бы измерять как темпы прироста русскоязычного сегмента Сети, так и основные количественные и качественные показатели поисковых машин.

Для оценки объема Рунета и определения величины русскоязычных архивов крупнейших поисковиков Илья оригинально модифицировал уже известный в то время подход Бхарата-Бродера [13], придав ему простоту и наглядность [11]. При определении доли выдачи какого-либо поисковика в выдаче других систем (основной пункт алгоритма Бхарата-Бродера) Илья предложил использовать небольшой массив из 120 редких однословных запросов, для которых выдача Яндекса не превышала 500 документов. Предполагалось, что в этом случае поисковики отключают все дополнительные фильтры и выдают полный набор документов из своего архива. Запросы отправлялись ко всем крупным поисковым системам, а результаты сравнивались с выдачей Яндекса и друг с другом. Подход Ильи опирался на гипотезу (основанную на многолетнем опыте разработки системы Яндекс) о пропорциональной зависимости между полными объемами архивов поисковиков и репрезентативными выборками из этих архивов.

Оказалось, что доля каждой поисковой системы в выдаче других поисковиков примерно одинакова, что доказывало независимость алгоритмов построения индексов этими системами. Тогда для определения объема Рунета нужно было разделить реальный размер базы Яндекса на его долю в выдаче, а для определения размера архива любого другого поисковика требовалось умножить размер архива Яндекса на долю этого поисковика. Просто и красиво, а главное, довольно точно, как показали независимые измерения.

Для измерения качества архивов Илья предложил уже полностью свои, оригинальные показатели «чистоты» (доля дубликатов в выдаче) и «свежести» (доля устаревших ссылок в выдаче). При определении «чистоты» использовалась (также придуманная Ильей для борьбы с почтовым спамом [11]) весьма остроумная методика «логарифмических шинглов». С ее помощью среди

полученных по редким запросам документов находились дубликаты, процент которых и характеризовал «чистоту» выдачи. При вычислении «свежести» делалась попытка найти основу слова редкого запроса в выдачах поисковиков. Процент неудачных попыток и определял «свежесть» архивов.

Можно, кажется, без конца вспоминать о лингвистических задачах, которыми интересовался Илья. Здесь мысли и об автоматической расстановке знаков препинания, и об определении морфемной структуры многокоренных слов, и о «лингвистических» расстояниях между частями речи и падежами и многие, многие другие. Но, как говорили древние, *sed satis verborum est*.

Подводя итог, можно сказать, что Илья был и душой, и мозговым центром практически всех лингвистических технологий Яндекса. Проекты, которые он начинал и которыми руководил, стали к настоящему времени мощными инструментами автоматической обработки ЕЯ-текстов, помогающими Яндексу находиться в числе мировых лидеров в области информационных технологий.

Литература

1. Автопоэт Яндекса. http://blog.yandex.ru/post/73398/
2. *Белоногов Г. Г., Зеленков Ю. Г.* Алгоритм морфологического анализа русских слов. Вопросы информационной теории и практики, N 53, М., ВИНИТИ, 1985. — 156 с.
3. *Зеленков Ю., Сегалович И., Титов В.* Вероятностная модель снятия морфологической омонимии на основе нормализующих подстановок и позиций соседних слов // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог'2005». — М.: Наука, 2005. 616 с.
4. *Зеленков Ю., Сегалович И.* Сравнительный анализ методов определения нечетких дубликатов для Web-документов. // Труды девятой всероссийской научной конференции RCDL'2007. Переславль-Залесский, 15–18 октября 2007.
5. Интернет-математика 2005. Автоматическая обработка веб-данных. М., 2005. — 504 с.
6. Интернет-математика 2007. Сб. работ участников конкурса. Екатеринбург, Изд-во Урал. ун-та, 2007. — 224 с.
7. *Кристофер Д. Маннинг* и др. Введение в информационный поиск. М., Вильямс, 2011. — 528 с.
8. Лингвистическое обеспечение системы ЭТАП-2. М.,1989.
9. Национальный корпус русского языка. http://www.ruscorpora.ru/
10. *Сегалович И., Маслов М.* Русский морфологический анализ и синтез с генерацией моделей словоизменения для не описанных в словаре слов. // Компьютерная лингвистика и интеллектуальные технологии: Труды международной конференции «Диалог'99». Т. 2. С. 547–552. Казань, 1998.
11. *Сегалович И., Зеленков Ю., Нагорнов Д.* Методы сравнительного анализа современных поисковых систем и определения объема Рунета. // Труды восьмой всероссийской научной конференции RCDL'2006. Суздаль, 17–19 октября 2006.

12. *Сичинава Д. В.* «Национальный корпус русского языка: очерк предыстории» (2005). http://www.ruscorpora.ru/sbornik2005/03sitch.pdf
13. *Bharat K. and Broder A.* A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. Proc. of the 7th International World Wide Web Conference, April 1998.
14. *Ilyinsky S., Kuzmin M., Melkov A., Segalovich I.* An efficient method to detect duplicates of Web documents with the use of inverted index. WWW'2002 — Eleventh International World Wide Web Conference.