	Language and Computers Topic 2: Searching	Outline	Language and Computers Topic 2: Searching	Searching	Language and Computers Topic 2: Searching
Language and Computers (Ling 384) Topic 2: Searching Adriane Boyd* Department of Linguistics, OSU Autumn 2005	Introduction Total Speech Searching in a Library Catalog Speech Characters Operators Searching the web Operators Improving searching Ranking of results Advanced search results Advanced searches with regular expressions System of regular force, he searche for using one of the control of the control of the control System of regular to the control of the control Total Components The components Th	Introduction Searching in a Library Catalog Searching the web Advanced searches with regular expressions	Introduction Test Speech Speech Searching in a Library Catalog Special characters Operation Searching the web Operation Intercoing searching Ranking of results Evaluating search results Advanced searches with regular expressions Syntax of regular expressions Intercoin and Searching Them	 ➤ A breathtaking number of information resources are available: books, databases, the web, newspapers, ➤ To locate relevant information, we need to be able to search these resources, which often are written texts: ➤ Searching in a library catalog (e.g., using OSCAR) ➤ Searching the web (e.g., using Google) ➤ Advanced searching in text corpora (using regular expressions) (e.g., using Opus) 	Introduction Best Speach Speach Speaching in a Library Catalog Speaid dwarders Operators Searching the web Operators Improving searching Ranking of results Evoluting search results Evoluting search results Advanced searches With regular expressions Syntax of regular expressions Cree, An example for using the compon and searching there
* The course was created by Markus Dickinson, Detmar Meurers and Chris Brew.					
Searching in speech	Language and Computers Topic 2: Searching	Searching in a library catalog	2/33 Language and Computers Topic 2: Searching	Basic searching in OSCAR	Language and Computers Topic 2: Searching
 One might also want to search for speech, e.g., to find a particular sentence spoken in an interview one only has a recording (audio file) of. With current technology, this is only possible if the interview is transcribed, using the IPA or another writing system. It is, however, already possible to detect the language of a spoken conversation, e.g., when listening in to a telephone conversation detect a new topic being started in a conversation In the following, we focus on searching in text. 	Introduction Text Searching in a Library Catalog Special characters Operators Searching the web Operators Improving searching Ranking of results Advanced searches with regular expressions Syntax of regular expressions Syntax of regular expressions Syntax of regular expressions Interpretable of the plant of the plant for the plant of the plant for the	 To find articles, books, and other library holdings, a library generally provides a database containing information on its holdings. OSCAR is the database frontend providing access to the library database at OSU. OSCAR makes it possible to search for the occurrence of literal strings occurring in the author, title, call number, etc. associated with an item held by the library. 	Introduction Text Speech Ing in a Library Catalog Speeds therecters Operators Searching the web Operators Herpoving searching Ranking of results Advanced search results Advanced search regular expressions Syntax of regular expressions Interpretable of the syntax o	 Literal strings are composed of characters which naturally must be in the same character encoding system (e.g. ASCII, ISO8859-1, UTF-8) as the strings encoded in the database. For literal strings, OSCAR does not distinguish between upper and lower-case letters (i.e. they aren't so literal after all ;-) Adjacent words are searched as a phrase. art therapy vitamin c 	Introduction Test Speech Speech Speech Speech Searching in a Library Catalog Special characters Operators Searching the web Operators Improving searching Ranking of results Evaluating search results Advanced searches With regular expressions Syntax of regular expressions Greg. An example for using regular expressions In the control of
 Keyword searching in OSCAR ▶ In addition to querying literal strings, the keyword search query language of OSCAR also supports the use of ▶ special characters to abbreviate multiple options ▶ special operators for combining two query strings 	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalog Special characters Operators Heroving searching the web Operators Heroving searching desurts Advanced searchess Advanced searchess	OSCAR: Special characters • Use * for 1–5 characters at end or within a word. • art* finds arts, artists, artistic • gentle*n • Use ** for any number of characters at end of word. art** finds artificial, artillery	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalog Speech Greators Searching the web Operators Introduction Searching of results Evaluating searching of results Evaluating search results Advanced searches	OSCAR: Literal Strings and Operators (I) • Use and or or to specify multiple words in any field, any order. • art and therapy • art or therapy • c+ or c++	Language and Computers Topic 2: Searching Introduction Test Speech Searching in a Library Catalog Speech Searching the web Operature Improving searching Ranking of results Evaluating searching Advanced searchessls Advanced searches
(boolean operators) or modifying the meaning of a single string (unary operators)	with regular expressions system of regular expressions system of regular expressions. System of regular expressions regular expressions. That corpora and searching them	 Use ? for a single character at end or within a word. gent1em?n The special * and ? characters must have at least 2 characters to their left. (→ for efficiency reasons) 	with regular expressions expressions syntax of regular expressions. Syntax of regular expressions complete repressions. The component of the c	► Use and not to exclude words. art and not therapy	with regular expressions Syntax of regular expressions Syntax of regular expressions Grey. An example for using Text coppor and searching them.

Use parentheses to group words together when using more than one operator. art therapy and not ((music or dance) ▶ Use near to specify words within 10 words of each other, in any order. ► art near therapy ▶ Use within n to specify words within n words of each

- other. The value of n has no limit.
 - ▶ art within 12 therapy

Searching the web Topic 2: Searchin

Searching in a

Library Catalog

Searching the web

Advanced searche

Grep: An example for using

Topic 2: Searchi

Introduction

Searching in a

Library Catalog

Searching the we

Advanced searche

Grep: An example for using regular expressions

Topic 2: Searchin

Introduction

Searching in a

Special characters

Operators

Ranking of results

with regular

expressions

Evaluating search result

Advanced searche

Grep: An example for usin

Library Catalog

Searching the web

Ranking of results

with regular expressions

with regular

expressions

Syntax of regular

A computer user

- wants to find something on "the web", i.e., in files accessible via the hypertext transfer protocol (http) protocol on the internet
- goes to a search engine = program that matches documents to a user's search requests
- enters a query = request for information
- gets a list of websites that might be relevant to the query
- evaluates the results: either picks a website with the information looked for or reformulates the query

The nature of the web Topic 2: Searching

Searching in a

Library Catalog

Searching the we

Advanced searche

Grep: An example for using

Text corpora and search

Language and Computers

Topic 2: Searchin

Introduction

Searching in a

Library Catalog

Searching the web

Advanced searches

with regular

Syntax of regular e

Text corpora and search

Topic 2: Searchin

Searching in a

Library Catalog

Searching the web

Evaluating search result

Grep: An example for using

with regular

expressions

with regular

expressions

Syntax of regular

► Web pages are generally less structured than a record in a library database (with title, author, subject, and other fields).

- ► One generally searches for words found anywhere in the document.
- ▶ It is, however, possible to include **metadata** in a web
- ▶ Metadata is additional, structured information that is not shown in the web page itself: e.g., the language a web page is in, its character encoding, author, keywords, etc.
- ► Example for a **meta tag**: <META name="keywords" lang="en-us" content="vacation, Greece">

Language and

Topic 2: Searching

Searching in a

Library Catalog

Searching the

Advanced searche

with regular

Syntax of regular

Search engines

OSCAR: Operators (II)

- Search engines (e.g., Google)
 - store a copy of all web pages
 - reate an index to provide efficient access to this large number of pages (e.g., Google currently searches over 4 billion pages)
 - compute a rank for each web page to be able to rank the query results
- Search engines differ in various ways:
 - stemming: treat bird and birds as the same or not
 - capitalization: treat trip and Trip the same or not
 - use of operators
 - special interface for advanced searching
 - how search results are ranked
 - · clustering: group similar results or not

Language and Computers Google: Operators (I)

- +: Require a word to occur in the result e.g., To find a restaurant that serves both tofu and BBQ one could try +tofu +BBO.
- -: Disallow a word from occurring in the result e.g., As a potatos purist, I search for potatos -potatoes
- ": Include synonyms of the word
- Quotation Marks (phrases) e.g., "What Cheer" when looking for sites on What Cheer, Iowa

Google: Operators (II)

intitle: Find words used in a title

- e.g., intitle: Buckeye finds only web pages which has this word in the title
- inurl: Find words used in the url
 - e.g., inurl:ling returns more linguistics webpages than ling does
- link: Find pages that link to a certain page
 - e.g., link:www.osu.edu to show pages linking to the main osu web page
- site: Find pages that are part of a single domain
 - e.g., I want to find strange attractions involving fish. Knowing one site which has such stuff, one can try fish site:www.roadsideamerica.com.

Language and Topic 2: Searching

Introduction

Searching in a Library Catalog

Searching the web Operators

Advanced searches

with regular

Google: Advanced searching

More elaborate web forms are provided as alternative to using operators:

- match all: matches all terms in your query
- ▶ match any: matches as many terms in your query as it e.g., I'm looking for a restaurant that has bbq or bb-que
- or barbeque in the title ⇒ most search engines return "match all" followed by "match any" results
- exclude: eliminate documents which contain certain words

Improving searching (I)

How can I make my searches better?

- ▶ Be on the watch for **ambiguity** = one word has multiple
- e.g., bed: flower bed, sleeping bed, truck bed
- Use synonyms and other related words e.g., plant: building, complex, works, power (distinguish
- ▶ Be aware of **stop words** = words that search engines ignore because they are "uninformative," such as the,
- e.g., The Police won't help you find the rock band any more than Police will

Improving searches (II)

► Exclude problematic words

- e.g., "jefferson airplane -starship" (if you don't want info on the Starship years)
- ▶ Be aware of parts of speech and what other guises they come in.
 - e.g., plant: planting, planter, planted (distinguish from power plant)
- ► Continually narrow your focus (using the feedback) e.g., Want to find information on the game Hearts
 - 1. hearts: too vague, too many non-card game sites → add a related word
 - 2. hearts cards: better, but still greeting cards listed → I see trick listed on one site's description and realize this makes for a good keyword
 - 3. hearts cards trick: good, but now we get card tricks → time for boolean expressions

Language and

Introduction

Searching in a Library Catalog

Searching the web

Evaluating search result with regular expressions

Grep: An example for using

* Ideally, the webpages matching a query are returned as an ordered fat based on a page selevance. * Ideally, the webpages matching a query are returned as an ordered fat based on a page selevance. * Focus can a source and a particular page of the selection o						
desemble how popular a page is (As a result, uppopular or now pages retailed now specific query to be found.) I Medily, the webpages matching a query are returned as an ordered late based on a pages relevance. I More an asserted manne, which does not understand language, determine the relevance of a particular page? I wow can a search engine, which does not understand language, determine the relevance of a particular page? I wow and search engine, which does not understand language, determine the relevance of a particular page? I wow and search engine, which does not understand language, determine the relevance of a particular page? I wow and search engine, which does not understand language, determine the relevance of a particular page? I wow and search engine, which does not understand language, determine the relevance of a particular page? I would be a page is relevance. I would be a page is relevance. I would be a page is relevance. I was the bot described the control of the date of inst to one page. I consuming him work that was circled on the page. I consuming him workers a web results was circled on the page. I consuming him workers a web results was circled on the page. I would be the control of the date of institution of the page is not to the end in the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page of the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the page on the topic of which are controlled to the	Ranking of results	Computers	Information used to rank results	Computers	Evaluating search results	Computers
Motivating regular expressions If one wants to be able to describe more complex patterns of words and text, sometimes boolean expressions aren't errough: In a large document I want to find addressess with a 2 zip code starting with 911 (around Pasadena, CA); but clearly we would not want to report back all occurrences of emergency phone numbers in the document. I want to find all ose uenall addresses which occur in a long text. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I would be nice to have a compact way of representing all of these options. Anything where you have to match a complex pattern so-called regular expressions are useful. The syntax of regular expressions (I) Regular expressions consist of strings of literal characters: c, 4166, natural language text. **Counters** **C	an ordered list based on a page's relevance.How can a search engine, which does not understand	Text Speech Spee	determine how popular a page is. (As a result, unpopular or new pages require a more specific query to be found.) Keeping track of the nature of links to a page; linked pages might be thematically related. e.g., Even if I never mention Sinclair Lewis on a page describing his book Babbit, it can be identified if many Sinclair Lewis sites link to my page. bonuses/penalties for sites known to be of high/low quality looking for keywords in metadata counting how often a web result was clicked on by a user (click-through measurement)	Text Speaching in a Library Catalog Special characters Operators Connectors Searching the web Operators Insproving searching Reading of results Evaluating search results Advanced Searches with regular expressions Syntax of regular expressions Care, An example to training team of the searching Text of regular expressions	query is? • precision: How many of the pages returned are the ones we want? e.g., Google gives me 400 hits for a query, 200 of which are related to the topic I want; precision = 50%. • recall: How many pages on the topic we wanted were actually given? (hard to calculate for web searching) e.g., Google gave me 200 pages I wanted, but there were actually 1000 pages on that topic out there somewhere on the internet; recall = 20%. We saw earlier how to use our initial results to refine our	Test Speach Searching in a Library Catalog Searching in a Library Catalog Speach drownsters Operators Searching the web Operators Intervine searching the web Operators Review of results Eximized or results
If one wants to be able to describe more complex patterns of words and text, sometimes boolean expressions aren't enough: In a large document I want to find addresses with a zip code starting with 911 (around Peasdena, CA); but clearly we would not want to report back all occurrences of emergency phone numbers in the document. I want to find aldresses with coccur in a long text. I want to find all uses email addresses with coccur in a long text. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5; for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. I'm writing an online fill-in-the-blank quiz, and I ask you to the same number of bs. I'm writing an online fill-in-the-blank quiz, and I ask you to the fill-in-the virting an work of the regular expressions as such have no linguistic contents, but they can be used to refer to strings encoding a natural language text. I'm writing an onlin	Motivating regular expressions	Language and	Regular expressions: What they are	Language and	Regular expressions: Tools that use them	Language and
The syntax of regular expressions (I) Language and Computers Topic 2: Searching Introduction That Injury Strings of literal characters: c, A100, natural Language, 30 years! Administrator classes: [Tt]he, bec [oa]me Finance: A-Z] (any capital letter) Administrator concerns to the surface of searches Finance: A-Z] (any capital letter) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions (III) Language and Computers Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching The syntax of regular expressions Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching Introduction That Injury The syntax of regular expressions Topic 2: Searching Injury The syntax of regular expressions Topic 2: Searching Injury The syntax of regular expressions Topic 2: Searching The syntax of regular expressions Topic 2: Searching	If one wants to be able to describe more complex patterns of words and text, sometimes boolean expressions aren't enough: In a large document I want to find addresses with a zip code starting with 911 (around Pasadena, CA); but clearly we would not want to report back all occurrences of emergency phone numbers in the document. I want to find all osu email addresses which occur in a long text. I'm writing an online fill-in-the-blank quiz, and I ask you to name the Jackson 5: for Jermaine, I want to accept Germaine, Germane, Jermain, and so on. It would be nice to have a compact way of representing all of these options. Anything where you have to match a complex pattern	Topic 2: Searching Introduction Teat Speach Searching in a Library Catalog Speach Improving searching Teaching of results Evaluating search results Advanced Searches with regular expressions Syrtax of regular expressions Sprack of regular expressions Instruction of the searching them	 A regular expression is a compact description of a set of strings, i.e., a language (in formal language theory). They can be used to search for occurrences of these strings Regular expressions can only describe so-called regular languages. This means that some patterns cannot be specified using regular expressions, e.g., finding a string containing any number of as followed by exactly the same number of bs. Note that just like any other formalism, regular expressions as such have no linguistic contents, but they can be used to refer to strings encoding a natural 	Topic 2: Searching Introduction Text Speach Speach Character Speach Speach Character Operators Searching the web Operators Searching the web Operators Expression of results Exhaulting search results Exhaulting search results Advanced Searches with regular expressions Syntax of regular expressions Syntax of regular expressions Text on control of the	 A variety of unix tools (grep, sed,), editors (emacs,), and programming languages (perl, python,) incorporate regular expressions. Implementations are very efficient so that large text files can be searched quickly; but not efficient enough for web searching → no web search engine offers them (yet). The various tools and languages differ w.r.t. the exact 	Topic 2: Searching Introduction Its Speech Searching in a Library Catalog Special characters Operators Searching the web Operators Intervine searching Reading of results Evaluating search results Advanced Searches with regular seyressions Syntax of regular expressions Syntax of regular expressions Teat coppon and searching them
► strings of literal characters: c, A100, natural language, 30 years! ► disjunction: • ordinary disjunction: devoured ate, famil(y ies) • character classes: [Tt]he, bec[oa]me • ranges: [A-Z] (any capital letter) • pegation: • pegation: • strings of literal characters: c, A100, natural Library Catalog Septim Catalog Septim Counters - optionality: ? Colou?r - optionality: ? Colou?r - colou?r - vany number of occurrences: * (Kleene star) - (0-9)* years - counters - colou?r - vany number of occurrences: * (Kleene star) - (0-9)* years - counters - colou?r - vany number of occurrences: * (Kleene star) - (0-9)* years - (0-9)* years - (0-9)* years - (0-9)* dollars - (0-	The syntax of regular expressions (I)	Computers Topic 2: Searching	The syntax of regular expressions (II)	Language and Computers Topic 2: Searching	The syntax of regular expressions (III)	Language and Computers Topic 2: Searching
[^a] (any symbol but a) [^a-Z0-9] (not an uppercase letter or number) beg.n for any character in between beg and n beg.n for any character in between beg and n the copons and searching from the copons and searchin	 strings of literal characters: c, A100, natural language, 30 years! disjunction: ordinary disjunction: devoured ate, famil(y ies) character classes: [Tt]he, bec[oa]me ranges: [A-Z] (any capital letter) negation: [a] (any symbol but a) 	Library Catalog Special characters Operators Searching the web Operators Improving searching Ranking of results Evaluating search results Advanced searches with regular expressions Syntax of regular expressions Grep. An example for using regular expressions	 optionality: ? colou?r any number of occurrences: * (Kleene star) [0-9]* years at least one occurrence: + [0-9]+ dollars wildcard for any character: . 	Library Catalog Special characters Operators Searching the web Operators Improving searching Ranking of results Evaluating search results Advanced searches with regular expressions Syntax of regular expressions Gray, in example for using regular expressions	special meaning (*, +, ?, (,), , [,]) it is preceded by a backslash (\) e.g., a period is expressed as _ • Operator precedence, from highest to lowest: parentheses () counters * + ? character sequences	Library Catalog Special characters Operators Searching the web Operators Ingroving searching Ranking of results Evaluating search results Advanced searchess with regular expressions System of regular expressions Grape. An example for using regular expressions Text copion and searching
25/33 26/33 27/33		25/33		26/33		27/33

 ▶ grep is a powerful and efficient program for searching in text files using regular expressions. ▶ It is standard on Unix, Linux, and Mac OSX, and there also are various ports to Windows (e.g., http://gnuwin32.sourceforge.net/packages/grep.htm, http://www.interlog.com/*tcharron/grep.html or http://www.wingrep.com/). ▶ The version of grep that supports the full set of operators mentioned above is generally called egrep (for extended grep). 	Language and Computers Topic 2: Searching Introduction Text Speech Speec	Grep: Examples for using regular expressions (I) In the following, we assume a text file f.txt containing, among others, the strings that we mention as matching. • Strings of literal characters: egrep 'and' f.txt matches and, Ayn Rand, Candy and so on • Character classes: egrep 'the year [0-9][0-9][0-9]' f.txt matches the year 1776, the year 1812, the year 2001, and so on • Escaped characters: egrep 'why\?' f.txt matches why?, whereas egrep 'why?' f.txt matches why and wh	Language and Computers Topic 2: Searching Introduction Test Speech Speech In Speech Speech Speech In Library Catalog Speech Coperators Searching the web Operators Introduction Searching Assistance of results Advanced searches with regular expressions Syntax of regular expressions Syntax of regular expressions Syntax of regular expressions Syntax of regular expressions Institute of regular expressions Syntax of regular expressions Institute of regular expressions Syntax of regular expressions Institute of regular expressions Institute of regular expressions Syntax of regular expressions Institute of regular expressions I	Grep: Examples for using regular expressions (II) • disjunction (): egrep 'G g' f.txt matches G or g, so egrep 'G gouda' f.txt matches gouda or Gouda. Note that (G g)ouda has the same effect. • grouping with parentheses: egrep 'un(interest excit)ing' f.txt matches uninteresting or unexciting. • Any character (.): egrep 'o.e' f.txt matches ore, one, ole	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalog Speech Constitute of Constitute Searching in a Library Catalog Speech Constitute Searching the web Operators Improving searching Ranking of results Feathbasing searching Ranking of results Advanced searches with regular expressions Syntax of regular expressions Greep. An example for uning Inspirer expressions Topic of the Constitute of
Grep: Examples for using regular expressions (III) * Kleene star (*): egrep 'a*rgh' f.txt matches argh, aargh, aaargh egrep 'sha(la)*' f.txt matches sha, shala, shalala, or if you're Van Morrison shalalalalalalala * One or more (+): egrep 'john+y' f.txt matches johny, johnny,, but not johy * Optionality (?): egrep 'joh?n' f.txt matches jon and john	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalog Speech characters Operators Searching the web Operators Improving searching the web Computer Residency of results Advanced searches with regular expressions Syntax of regular expressions Syntax of regular expressions Text corpora and searching them	A corpus is a collection of text. Corpora with the works of various writers, newspaper texts, etc. have been collected and electronically encoded. Corpora can be quite large The British National Corpus is a 100 million word collection representing a wide cross-section of current written and spoken British English. Another example is the European Parliament Proceedings Parallel Corpus 1996–2003.	Language and Computers Topic 2: Searching Introduction Test Speech Searching in a Library Catalog Speedi characters Operation Improving searching the web Operation Improving searching are search Resident of results Advanced search results Advanced searches with regular expressions Syntax of regular expressions Graps: An example for using regular expressions Inter corpora and searching Test compone and searching Test compone and searching Test compone and searching	Both the BNC and the European Parliament corpus can be searched using on-line web-forms. Both of the web forms allow regular expressions for advanced searching. To provide efficient searching in large corpora, in these search engines regular expressions over characters are limited to single tokens (i.e. generally words). BNC:	Language and Computers Topic 2: Searching Introduction Text Speech Searching in a Library Catalog Special characters Operators Horrowing searching the web Operators Improving searching the web Coperators Improving searching Ranking of results Evaluating search results Advanced searches with regular expressions Syntax of regular expressions Gree; An example for using regular expressions These compensations are searches and searching