Exploring synonyms within large commercial site search engine queries

Julia Kiseleva, Andrey Simanovsky

HP Laboratories
HPL-2011-41

Keyword(s):

synonym mining, query log analysis

Abstract:

We describe results of experiments of extract-ing synonyms from large commercial site search engine query log. Our primary object is product search queries. The resulting dictionary of synonyms can be plugged into a search engine in order to improve search results quality. We use product database to extend the dictionary.

External Posting Date: April 6, 2011 [Fulltext] Approved for External Publication
Internal Posting Date: April 6, 2011 [Fulltext]

Exploring synonyms within large commercial site search engine queries

Julia Kiseleva, Andrey Simanovsky

HP Labs Russia

Abstract. We describe results of experiments of extracting synonyms from large commercial site search engine query log. Our primary object is product search queries. The resulting dictionary of synonyms can be plugged into a search engine in order to improve search results quality. We use product database to extend the dictionary.

Keywords: synonym mining, query log analysis

1 Introduction

A large commercial site is an information portal for customers where they can find everything about the vendor's products e.g. manuals, drivers, etc. A large commercial site has a search engine and its main function is to help customers to retrieve appropriate information. We regard a user search query as a product query when the user's intent is to retrieve information about hp products and services including manuals, drivers, and support.

One way to improve search quality is to utilize a dictionary of synonyms that incorporates a document collection vocabulary and vocabulary of different users. We analyze incoming queries for synonym terms which could be included into a thesaurus. We attempted several techniques in order to detect synonymous terms and queries among queries from a large commercial site search engine.

Query expansion is defined as a stage of the information retrieval process during which a user's initial query statement is extended with additional search terms in order to improve retrieval performance. Query expansion is rationalized by the fact that initial query formulation does not always reflect the exact information need of a user. The application of thesauri to query expansion and reformulation has become an area of increasing interest.

Three types of query expansion are discussed in the literature: manual, automatic, and interactive (including semiautomatic, user-mediated, and user-assisted). These approaches use different sources of search terms and a variety of expansion techniques. Manual approach does not include any knowledge about a collection while interactive approach implies a query modification by a feedback process. However, assistance could be sought from other sources, including a dictionary or a thesaurus. In the query expansion research, one of the biggest issues is to generate appropriate keywords that represent the user's intention.

Spelling correction [1] is also related to synonyms detection as its techniques are applicable, especially for product synonyms which share a lot of common words.

The methods described above have used only a query set as input data. But there are a few published approaches which use external data sources for synonym detection to make the technique more robust.

The goal of this project is detecting synonymous terms in search queries which are submitted by users to a large commercial site search engine. We also provide recommendations for enhancing search quality on the large commercial site.

The remainder of the report is organized as follows. We review related work in Section 2. The problem is formulated in Section 3. Section 4 discusses algorithms that were utilized; in particular, Sections 4.1 - 4.2 present functions we use as measures of similarity. Section 5 describes our experimental data set and compares experimental results. Finally, Section 6 summarizes our contribution.

2 Related Work

There are a lot of papers related to synonym detection in search queries. Thesauri have been recognized as a useful source for enhancing search-term selection for query formulation and expansion [4], [5]. Terminological assistance may be provided through inclusion of thesauri and classification schemes into the IR system.

In a series of experiments on designing interfaces to the Okapi search engine it was found that both implicit and explicit use of a thesaurus during automatic and interactive query expansion were beneficial. It was also suggested that while the system could find useful thesaurus terms through an automatic query-expansion process, terms explicitly selected by users are of particular value ([4], [6]).

The paper [3] presents a new approach to query expansion. Authors proposed *Related Word Extraction Algorithm* (RWEA). This algorithm extracts words from texts that are supposed to be strongly related to the initial query. RWEA weights were also used in *Robertson's Selection Value* (RSV), a well known method for relevance feedback [4], weighting scheme. Query expansion was performed based on the results of each method (RSV, RWEA, and RSV with RWEA weights) and a comparison was made. RWEA evaluates a word in a document and RSV evaluates a word among several documents, consequently, the combination should perform uniformly well. Experimental results corroborated that statement: the combined method works effectively for all queries on average. In particular, when a user inputs initial queries which results have Average Precision (AP) under 0.6 the method obtains the highest Mean Average Precision (MAP). It also obtains the highest among the three methods MAP on experiments with navigational queries. However, RWEA obtains the highest MAP on experiments with informational queries. Experimental results show that effectiveness of a method for query expansion depends on the type of queries.

There are a lot of research papers about query spelling correction [1] which were published recently. We think that this area is also related to synonym detection as its techniques are applicable. For example in [7] authors consider a new class of similarity functions between candidate strings and reference entities. These similarity functions are more accurate than previous string-based similarity functions because they aggregate evidence from multiple documents and exploit web search engines in order to measure similarity. They thoroughly evaluate techniques on real datasets and demonstrate their precision and efficiency.

In [2] authors present a study on clustering of synonym terms in search queries. The main idea is that if users click on the same web-page after submitting different search queries those queries are synonyms.

3 Problem Statement

Our goal is to build a thesaurus of synonyms terms which are related to respective products. We also provide a set of recommendations for enhancing quality of search results returned by the large commercial site search engine.

4 Algorithms

4.1 Similarity Distance Metrics

We perform experiments with token-based and term-based similarity metrics. We choose this metrics because their efficiency was proved in literature [11], [13].

4.1.1 Token-based distance

There are a lot of token-based string similarity metrics which are described in the literature.

Levenshtein distance (LD) is a measure of the similarity between two strings, which we will refer to as the source string (s) and the target string (t). The distance is the number of deletions, insertions, or substitutions required to transform s into t. For example,

- If s is "test" and t is "test", then *LD(s,t) = 0*, because no transformations are needed. The strings are already identical.
- If s is "test" and t is "tent", then *LD(s,t) = 1*, because one substitution (change "s" to "n") is sufficient to transform s into t.

The greater the Levenshtein distance is the more different are the strings. Levenshtein distance is also called *edit distance*.

Smith –Waterman distance [11] is similar to Levenshtein distance. It was developed to identify optimal alignments between related DNA and protein sequences. It has two parameters, a function d and a gap G. The function d is a function from an alphabet to cost values for substitutions. The gap G allows costs to be attributed to insert and delete operations. The similarity score D is computed with a dynamic programming algorithm described by the equation below:

$$D(i,j)=\max\begin{cases} 0 \, // \, start \\ D(i-1,j-1)-d(si,tj) \, // \, subst/copy \\ D(i-1)-G \, // \, insert \\ D(i,j-1)-G \, // \, delete \end{cases}$$

The final score is given by the highest valued cell. Table 1 presents the example of score calculation.

	C	O	H	E	N
M	0	0	0	0	0
C	2	1	0	0	0
C	2	1	0	0	0
O	1	4	3	2	1
H	0	3	6	5	3
N	0	2	5	5	7

Table 1. Smith-Waterman calculation between string "cohen" and "mccohn" where G = 1, d(c,c) =2, d(c,d) = +1.

Smith-Waterman-Gotoh [12] is an extension of Smith-Waterman distance that allows affine gaps within the sequence. The Affine Gap model includes variable gap costs typically based upon the length of the gap l (W_l). If two sequences, A $(=a_1\ a_2\ a_3\ \ldots\ a_n)$ and B $(=b_1\ b_2\ b_3\ \ldots\ b_m)$, are compared the formula for dynamic programming algorithm is: $D_{ij}=max\{D_{i-1, j-1} +d(a_i,b_j), max_k \{D_{i-k,j} -W_k\}, max_l \{D_{i, j-l} -W_l\}, 0\}$, where D_{ij} is in fact maximum similarity of two segments *ending* in a_i and b_j respectively.

Two affine gap costs are considered, a cost for starting a gap and a cost for continuation of a gap.

Definition: The taxicab distance, d_1, between two vectors p, q in an n-dimensional real vector space with fixed Cartesian coordinate system, is the sum of the lengths of the projections of the line segment between the points onto the coordinate axes:

$$d_1(p,q) = \| p - q \|_1 = \sum_{i=1}^{n} | p_i - q_i |,$$

Where $p = (p_1, p_2, \ldots, p_n)$ and $q = (q_1, q_2, \ldots, q_n)$ are the two vectors.

The taxicab metric is also known as **rectilinear distance**, **L_1 distance** or **ℓ_1 norm**, **city block distance**, **Manhattan distance**, or **Manhattan length**.

4.1.2 Term based distance

We choose cosine similarity metric as a term-based distance. Cosine similarity is a measure of similarity between two vectors which is equal to the cosine of the angle between them. The result of the Cosine function is equal to 1 when the vectors are collinear or between 0 and 1 otherwise.

Cosine of two vectors can be easily derived by using the Euclidean Dot Product formula:

$$a * b = || a |||| b || \cos \alpha$$

$$similarity = \cos(\alpha) = \frac{a * b}{|| a |||| b ||} = \frac{\sum_{i=1}^{n} a_i \times b_i}{\sqrt{\sum_{i=1}^{n} (a_i)^2} \times \sqrt{\sum_{i=1}^{n} (b_i)^2}}$$

As a weighting function we used a *tf*idf* weight. The *tf (term frequency)* in the given document is simply the number of times a given term appears in that document:

$$tf = \frac{n_{i\,j}}{\sum_k n_{k\,j}}$$

where $n_{i,j}$ is the number of occurrences of the considered term t_i in document d_j, and the denominator is the sum of number of occurrences of all terms in document d_j, that is, the size of the document $| d_j |$.

The *idf (inverse document frequency*) is a measure of the general importance of the term :

$$idf = \log \frac{| D |}{| \{ d : t_i \in d \} |}$$

We selected *tf*idf* weight. It combines two aspects of a word, the importance of word for document and its discriminative power within the whole collection.

Each query was regarded as a *document* in the collection. *Tf* is the frequency of a term in a query. It is almost always equal to 1 and *idf is* the ordinary inverse document frequency.

4.2 Probabilistic Model

4.2.1 Source Chanel Model

In paper [1] authors apply *source channel model* to the error correction task. We explore the possibility of applying it to finding synonyms. Source channel model has been widely used for spelling correction. Using source channel model, we try to solve an equivalent problem by applying Bayes’ rule and dropping the constant denominator:

$$c^{*} = argmax_{c \in C} P(q \mid c)P(c), where\ q\ is\ query, c\ is\ correction\ \ candidate.$$

In this approach, two components of generative model are involved: *P(c)* characterizes user’s intended query *c* and *P(q|c)* models error. The two components can be estimated independently.

The source model (*P(c)*) could be approximated with n-gram statistical language model. It is estimated with tokenized query logs in practice for multi-term query. Consider, for example, a bigram model. *c* is a correction candidate containing *n* terms, c= $c_1 c_2 .. c_n$, then *P(c)* could be written as a product of consecutive bigram probabilities:

$$P(c) = \prod P(c_i \mid c_{i-1})$$

Similarly, the error model probability of a query is decomposed into generation probabilities of individual terms which are assumed to be independent:

$$P(q \mid c) = \prod P(q_i \mid c_i)$$

Now the word synonymy can be accessed via correlation. There are different ways to estimate distributional similarity between two words, and the one we propose to use is *confusion probability*. Formally, confusion probability P_c estimates the possibility that a word w_1 could be replaced by another word w_2 [1]:

$$P_c(w_2 \mid w_1) = \sum_w P(w \mid w_1) \frac{P(w \mid w_2)}{P(w)} P(w_2) ,$$

where *w* belongs to the set of words that co-occur with both, w_1 and w_2.

For synonym detection we assume that w_1 is an initial word and w_2 is a synonym. Confusion probability $P_c(w_2 \mid w_1)$ models the probability of w_1 being rephrased as w_2 in query logs.

4.2.2 Utilizing database as external data container

As we mention in section “Related works”, there is a successful practice of utilizing external sources to discover synonyms. We present a novel method which makes use of a database with product names to enhance synonym detection estimated in the previous section. The database provides new ways to detect synonym terms because it contains product names which are related to the queries but could be expressed in other words. Synonym terms from the database are extremely useful for detecting related products during search process.

We introduce an analog of confusion probability between words in the query and terms in the database.

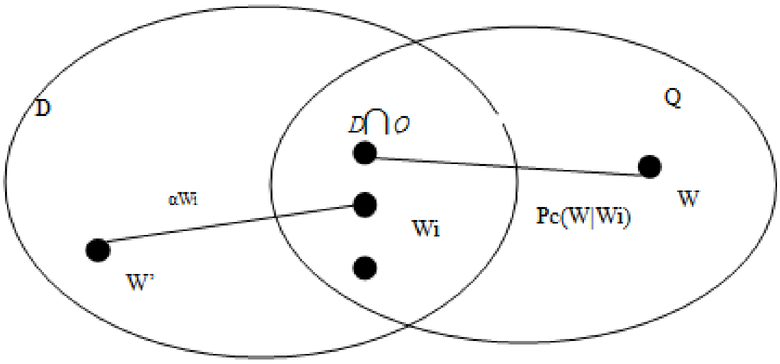


Figure 1. Metrics inside search query tokens and product names database

Figure 1 shows sets of tokens in a database (*D*) and in a query log (*Q*). $D \bigcap Q$ is an intersection of terms in the database and the query log; w_i is a token from the intersection. $P_c(w \mid w_i)$ is the confusion probability from [1].

α_{w_i} depicts a similarity function within the space of database terms between the term $w^{'}$ and the term w_i, which we choose to be Manhattan distance because it performed best as token-based similarity measure. { w_i } is a set of terms which occur in the intersection between database and queries (in $D \bigcap Q$).

We extend a notion of *confusion probability* between $w^{'}$ and *w* where *w* is term which occurs only in queries and $w^{'}$ is term which occurs only in the database.

We propose two ways of introducing confusion probability extension (in both formulas *i* indexes words of the intersection):

1. $\overline{P_{\max\_c}(w, w^{'})} = \max_i C_i = \max_i (P_c(w \mid w_i) * \alpha(w_i, w^{'}))$
2. $\overline{P_c(w, w^{'})} = \sum_{W_i} C_i P(w_i) = \sum \alpha(w_i, w) * P_c(w \mid w_i) * P(w_i)$

Note that the natural desired property $\overline{P_c(w,w')} = P_c(w|w')$ if $w' \in D \bigcap Q$ is not automatically met by the introduced extension.

Another possible approach to extend the confusion probability is to introduce $\overline{\overline{P_C(w,w'')}}$ according the joint distribution of (w_i, w_j), where $w_i \in D \bigcap Q$ and $w_j \in D \bigcap Q$.

5 Experiments

In order to perform initial data filtering, we have built basic statistics of the query log and found notable properties of the current large commercial site search engine traffic, which are presented in section 5.1. Next we evaluated the metrics presented in the Section 4. We present the evaluation in subsequent sections together with sample results.

5.1 Data Description

In this section we present data description and some statistics which will help us to understand *data nature*. By data nature we mean answers to the following questions:

- Where the queries have come from?
- What is the average length of a query?
- What is the list of stop words for the large commercial site search engine query log?

The query log used for analysis is collected during 8 days. It contains **189185** queries, **76972** unique queries, and **20261** queries which occur more than one time. The average length of the query is **2.008 words**. Table 2 provides a detailed query log description. The log does not contain any additional information about users except ip-addresses. They do not uniquely identify users.

Ip-address[1]	Time	Request	Browser information	Status	Status1	Return page
76.119.*.*	05/Jun/2010:00:00:00 +0000	GET /query.html?lang=en&search=++&qt=pavillion+6130+add+replace+expansion&la=en&cc=us&charset=utf-8 HTTP/1.1	Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.3) Gecko/20100401 Firefox/3.6.3	200	8200	http://www.hp.com/

Table 2. Query log description.

The query frequency distribution in the log is presented below, on the figure 2.

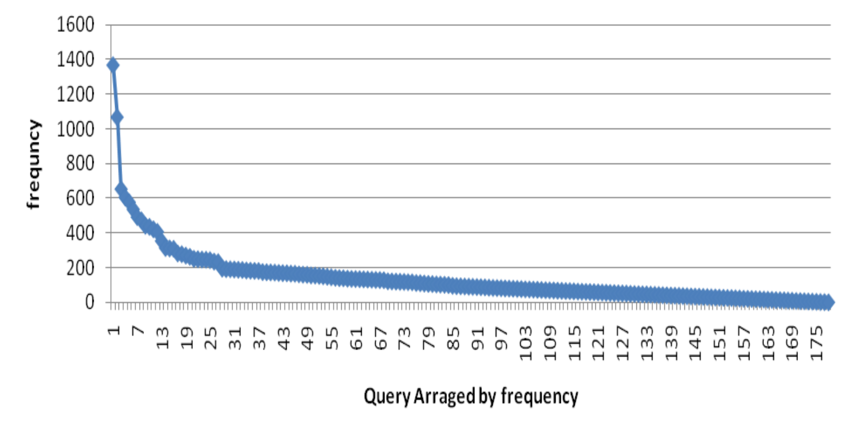


Figure 2. Query's frequency distribution.

Top most frequent queries are given in the Table 3. The most popular queries are non-product queries like "*google*". We think that those queries are most frequent because they have come from internal corporate users. Probably it happens because the commercial site page is by default a start page of company's employees.

[1] Here and further on IP addresses are partially obfuscated because of privacy considerations

Query	Frequency
search:	1066
Google	610
Drivers	579
hp officejet j4500 series search	535
Slate	439
hp deskjet f2200 series search	421
Warranty	363
hp business availability center	354
hp deskjet f4200 series search	246
Tablet	232
go instant	214

Table 3. The most frequent queries in the log

There is a parameter *web section* in the request that shows what category on site was selected by a user. From our point of view the query distribution by topic could be useful in order to understand user behavior. We built statistics by web section from query URLs. This web section is related to the query topic. The statics is demonstrated in the Table 4. The total number of web section queires is 527 which is 0.35 % of the whole number of queries, i.e. web section functionality is not popular with the users.

Web Section Topics	Frequency
small & medium business	153
Home	108
compaq.com	70
home & home office	55
home & home office section only	42
small & medium business site	37

hp procurer networking	27
products and services	10
home & home office only	9
hp promotions only	6
business technology optimization (bto) software	4
learn about supplies	3
hp online store	2
hp services	1
Total	**527 (0, 35%)**

Table 4. Distribution of web section queries

5.2 Data Preprocessing

5.2.1 Data Filtering

For some of the approaches that we apply, as well as to make distinction between external and internal use of the site, we need per user data. To obtain per-user statistics we develop a technique for data filtering.

We figure out that there were ip-addresses which send many requests to the search engine. We give examples of such ip-addresses, which had more than 1000 requests, in the Table 5. We believe that most of those search queries are sent from company's employees' computers through corporate proxies. The corporate ip-addresses are marked with bold in the Table 5. We called this set of ips *"non-confidential"* and they were removed from the data set.

Ip-address	Frequency
16.246.*.*	**6300**
16.247.*.*	**6068**
16.246.*.*	**3321**
16.246.*.*	**3313**
16.247.*.*	**3217**

16.247.*.*	**3168**
16.247.*.*	**3166**
16.246.*.*	**3112**
59.160.*.*	2802
221.134.*.*	2020
170.65.*.*	**1878**
170.65.*.*	**1200**

Table 5. Top “non-confidential” ip-addresses

We calculated statistics of requests from all ip-addresses and from non-confidential ip-addresses. The statistics are presented in the Table 6. We conclude that at least 25% of search queries originate from inside the company.

Date	**Number of confidential requests**	**Number of all requests**	**Delta**
1 June 2010	25199	31054	5855
2 June 2010	25709	32485	6776
3 June 2010	25382	32729	7347
4 June 2010	20504	26101	5597
5 June 2010	14336	18587	4251
6 June 2010	13004	16932	3928
7 June 2010	25485	31295	5810
8 June 2010	1	2	1
Total	**149620**	**189185**	**39565**

Table 6. Daily query statistics per origin

To make our methodology more robust we build a list of stop-words. It contains prepositions and term *‘hp’*. We used this list to clean up queries in the log.

5.2.2 Identification of user session time

In one session a user may try to pursue single information need and reformulate queries until he/she gets a desired result. Thus, analyzing user sessions in order to find synonymous queries seems reasonable.

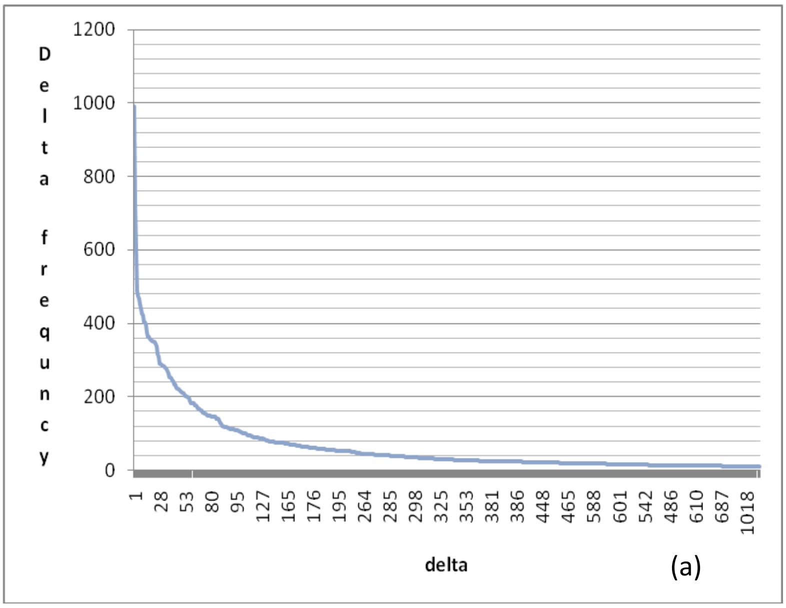
We filtered ip-addresses form the log according to the algorithm described in Section "Data filtering" to identify user session.

Definintion1: ***Delta*** is a time in seconds between two contiguous clicks from the same ip.

Definintion2: ***Delta frequency*** – frequency of delta in the whole query log.

For both cases, with non-confidential ip-addresses and without non-confidential ip-addresses, we built plots which are presented on Figure 3. We suppose that we should see how a user rephrases the query or expands it.

We used Manhattan Distance to find synonyms because it has performed well in previous experiments.



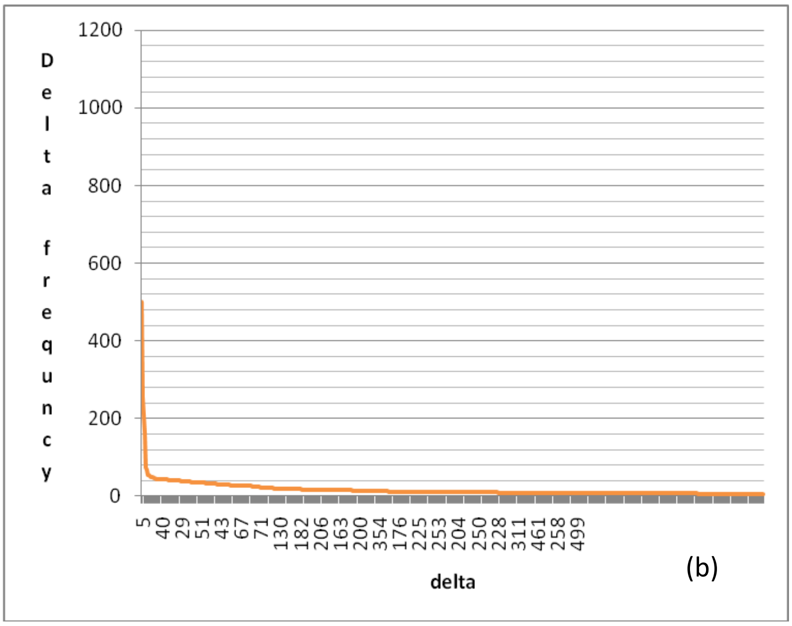


Figure 3. (a) a histogram of deltas which start from 5 seconds for all ip-addresses and (b) a histogram for deltas which start from 5 seconds for set of ip-addreses without non-confidential ips .

5.3 Evaluation Metrics

We use *precision* as an evaluation metric for our experiments. Its formula is given below:

$$\Pr ecision = \frac{\# correct\ \_\ results}{\# total\ \_\ results}$$

5.4 Experiments with different token based similarity metrics

The first approach that we considered for finding synonyms originates in the task of matching similar strings[2]. To characterize whether or not a candidate string is synonymous to another string, we compute the string similarity score between the candidate and the reference strings [10, 6].

[2] We use simmetrics library (http://staffwww.dcs.shef.ac.uk/people/S.Chapman/stringmetrics.html)

Unfortunately, there is no gold standard for evaluating synonyms discovery in query logs and we have to build ground truth. After performing experiments with different metrics we select top results and evaluate them manually.

We decided not to make general pooling and evaluate precision at 100 metric instead. We believe that top similar pairs are more stable that pairs similar to a given one. The results of evaluation and the volume of gold standard are presented in the Table 7.

Token-based Metric	**Gold Standard Size**	**Precision**
Levenshtein Distance	250	0.232
Smith-Waterman Distance	100	0.49
Smith-Waterman-Gotoh Distance	100	0.40
Manhattan Distance	100	0.88

Table 7. Results of experiments with proposed token-based metric.

Manhattan Distance shows the best precision at 100.

The main reason for low precision is that string similarity does not imply synonymy. E.g. strings “hp deskjet 960c” and “deskjet 932c” are similar according to similarity metric but they represent different models of printers and this is not a case of synonymy.

5.4.1 Synonyms detection by using click on the same URL

A hypothesis suggested in [2] claims that if users click on the same search result URL their queries should be *synonyms.* We explored that hypothesis on our data. The Table 8 shows a few examples that were obtained:

Id	**Queries from the same clicked url**		
1	hp deskjet 845c	hp deskjet d1360	
2	hp deskjet d1360	hp deskjet 845c	
3	hp laserjet 4350tn	hp laserjet 1102	
4	hp laserjet 1102	hp laserjet 4350tn	
5	hp photosmart c6380	hp photosmart a524	hp photosmart c4240

6	hp psc 1300	hp psc 1315	hp psc 2410
7	hp psc 1315	hp psc 1300	hp psc 2410
8	hp photosmart a524	hp photosmart c6380	hp photosmart c4240
9	hp photosmart c4240	hp photosmart c6380	hp photosmart a524
10	hp psc 2410	hp psc 1300	hp psc 1315
11	hp pavilion dv6500	hp pavilion dv2000	hp pavilion dv3

Table 8. Examples of synonyms through clicks on the same URL

One can see that we obtained low precision. A clue to that issue is that queries which contain different model numbers are regarded as synonyms. We expected that users will reformulate a query by replacing a term, but we found that users mostly replace a model number.

5.4.2 Synonyms detection by using user session

We perform experiments with the purpose of finding similar terms within the query session using the methodology to detect a user session that we described in section 5.2.2.

We evaluated 203 queries manually and this set is our gold standard for expert evaluation. We obtained precision equal to **0.63.**
A few examples of synonyms in one user session are given in the Table 9. The Table 9 also presents a similarity value between queries within the session.

User IP	**Query1**	**Query2**	**Similarity Value**
109.200.172.250	audio sp27792	sp27792	0.6666667
109.205.112.114	hpdv6-1153ei drivers	dv5-1153ei drivers	0.5
116.48.144.139	hp proliant ml350 g6	ml330 g6	0.4
116.48.144.139	ml330	ml330 g6	0.6666667

116.49.98.57	hp officejet j4500 series search	hp officejet j4500 series warranty registration	0.6666667

Table 9. Synonymous queries within a user session

5.5 Experiments with term based metric

We inflated weight for terms that are numbers or contain numbers. It was done in order to avoid regarding queries with different model numbers as synonyms. Candidate pairs of synonymous queries which had cosine similarity less than 0.7 were filtered. We have evaluated 150 queries and obtained precision of 0.4. Almost all results are synonyms expansion.

We did not include the term *"hp"* and prepositions into features space because we consider them as stop words.

A few examples of synonyms found with cosine similarity are presented in the Table 10. The obtained set of synonyms could be divided into two categories:

- query expansion (pairs 1, 2, and 3)
- query rephrasing (pair 4). In this case we can conclude that terms *'laptop'* and *'notebook'* are synonyms.

id	Initial Query	Query Synonym
1	410 photosmart	410
2	hp laserjet 4250n	4250n
3	rx3715	ipaq rx3715
4	4510 laptop	4510 notebook

Table 10. Examples of query synonyms obtained with cosine similarity metrics

5.6 Experiments with confusion probability

In this section we applied another approach to synonyms detection. This approach detects synonyms on the level of single words rather than whole queries and it recalls source channel model.

Some of the top results of the described synonyms detection method are presented in the Table 11. Most of presented synonyms could be characterized by following categories:

- paronymous terms like *'face'* and '*facial'* ;
- misspelling like *'Designerjet'* and *'Designjet'*;
- different forms of the same word like *'dv42160us'* and *'dv4-2164us'*.

Query term	**Query term should be similar**	**Confusion probability**
Designerjet	Designjet	0.75
Windows2008	2008	0.38
Twain	Twin	0.2
Michael	Micheal	0.148
dv42160us	dv4-2164us	0.564
Facial	Face	0.5625
Vitamine	Vitamin	0.2
Ms-6390	Ms6390	0.125
Technisch	Farm	0.375

Table 11. Synonymous terms in queries detected with confusion probability

6 Conclusion and recommendations

We discovered that all obtained synonyms can be classified into the following groups:

1. Misspellings.
2. Different forms of a word (mostly plural form)
3. Term and digit. Terms adhering the following regular expressions: *"Digit Space* Letter" and "Letter Space* Digit*.
4. Query expansions.
5. Rephrasings. It is the type of synonyms which is the most interesting for us.

The Table 12 contains examples of the above categories.

Category	**Initial Query**	**Synonyms Query**
Misspelling	1) alanta 2) laser 3) video 4) Designerjet	1) Atlanta 2) Leser 3) Video 4) Designerjet
Different form of the word	Warranties	Warranty
Term and digit	dv 8	dv8
Query expansion	hp office locations in india	hp india
Rephrasing	1) Remove 2) Activation 3) How to 4) Total care 5) Call center	1) Uninstall 2) Product key 3) Help, not working, support 4) Adviser 5) service center

Table 12. Synonyms categories with examples

According the discovered groups of synonyms we give the following recommendations:

1. Make spelling correction in run time. We can identify and store a list of most common misspelled terms. The appendix B demonstrates that currently search engine at the site cannot detect a misspelling. The Figure 5 shows that the search engine does not correct misspelling and returns irrelevant results.
   For now we cannot say that we have detected the whole list of misspellings because the current query log does not have enough data.
2. We think that storing different forms of terms will improve search quality.
3. Make data normalization. Terms adhering the following regular expressions:"*Digit Space* Letter" and "Letter Space* Digit* should be normalized. We should normalize incoming queries and data in the database.
   The appendix C contains two Figures, 7 and 8, which show how search result could change depending on form of writing for hard drive capacity.
4. We need more data to detect query expansions. The search engine has query reformulations service but sometimes very weird suggestions are returned. One of the examples is presented in appendix A, Figure 4. The site should have a product oriented search engine but suggested queries look like most frequent queries and are not related to products. An example could be found in the Appendix A, the Figures 5 and 6.
5. We present novel technique for synonym detection in this report. We need more data to detect strong list of rephrasing synonyms.

We detected two problems with data set:

- The majority of queries come from internal corporate users and they are not product search queries. We think that this peculiarity is not inherent to the specific query log and reflects general issues with the current search functionality on the site.
- Statistics of the one week log are not enough to detect strong synonym patterns. We total number of extracted synonym pairs counts on tens. We hope that a longer log can increase that number with close to linear dependence on the log size.

7 References

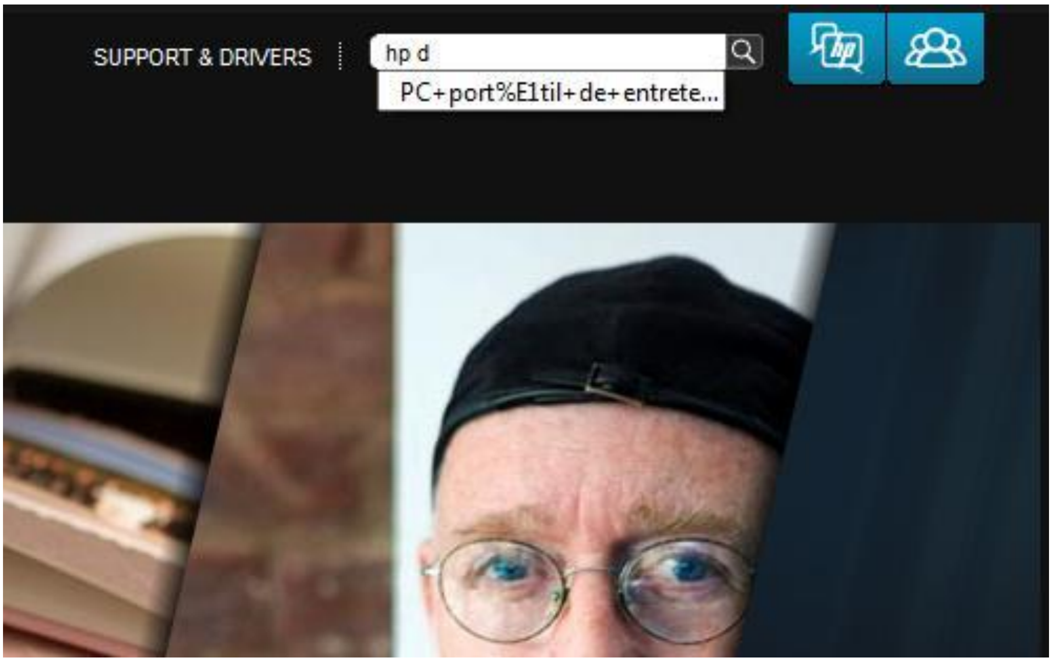
1. Mu Li, Muhua Zhu, Yang Zhang, Ming Zhou. Exploring Distributional Similarity Based Models Query Spelling Correction. *In processing of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 1025–1032, 2006
2. Jeonghee Yi, Farzin Maghoul.Query clustering using click-through graph. In processing of SIGIR, 2009
3. Tetsuya Oishi, Shunsuke Kuramoto, Tsunenori Mine, Ryuzo Hasegawa, Hiroshi Fujita, Miyuki Koshimura: A Method for Query Expansion Using the Related Word Extraction Algorithm. Web Intelligence/IAT Workshops 2008:41-44.
4. Beauliev, M. (1997). Experiments of interfaces to support query expansion Journal of Documentation, 53(1), 8–19.
5. Brajnik, G., Mizzaro, S., & Tasso, C. (1996, August). Evaluating user interfaces to information retrieval systems: A case study on user support. Proceedings of the 19th annual conference on Research and Development in Information Retrieval (ACM/SIGIR) (pp. 128–136). Zurich, Switzerland.
6. Jones, S., Gatford, M., Hancock-Beaulieu, M., Robertson, S.E.,Walker,W.,& Secker, J. (1995). Interactive thesaurus navigation: Intelligence rules Ok? Journal of the American Society for Information Science, 46(1), 52–59.
7. Surajit Chaudhuri, Venkatesh Ganti, Dong Xin. Exploiting Web Search to Generate Synonyms for Entities, WWW 2009
8. K. Chakrabarti, S. Chaudhuri, V. Ganti, and D. Xin. An efficient filter for approximate membership checking. In *SIGMOD Conference*, pages 805-818, 2008
9. W. W. Cohen and S. Sarawagi. Exploiting dictionaries in named entity extraction: combining semi-markovextraction processes and data integration methods. In*KDD*, pages 89-98, 2004
10. C. H. Bennett, P. Gács, M. Li, P. M. B. Vitányi, and W. Zurek, “Information distance,” *IEEE Trans. Inform. Theory*, vol. 44, pp. 1407–1423, July 1998.
11. Smith, T. F. and Waterman, M. S. “Identification of common molecular subsequences”, J. Mol. Biol., pp. 195-197, 1981

12. Gotoh, O. "An Improved Algorithm for Matching Biological Sequences". Journal of Molecular Biology. 162:705-708, 1981
13. Rishin Haldar, Debajyoti Mukhopadhyay.Levenshtein Distance Technique in Dictionary Lookup Methods: An Improved Approach. In *processing of CoRR* abs/1101.1232 (2011).

8 Appendix

8.1 A.

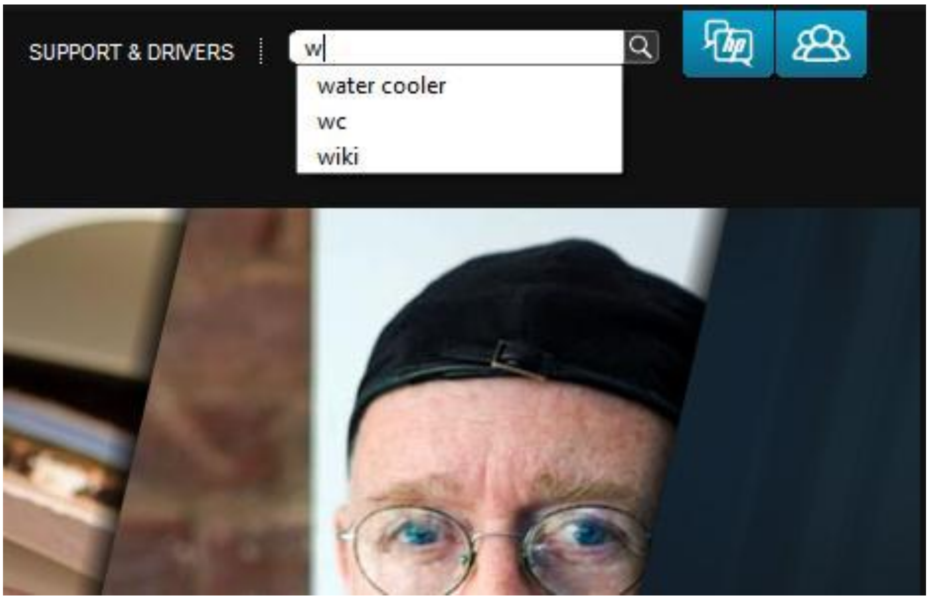
Figure 4. Controversial query suggestions:



SUPPORT & DRIVERS

hp d

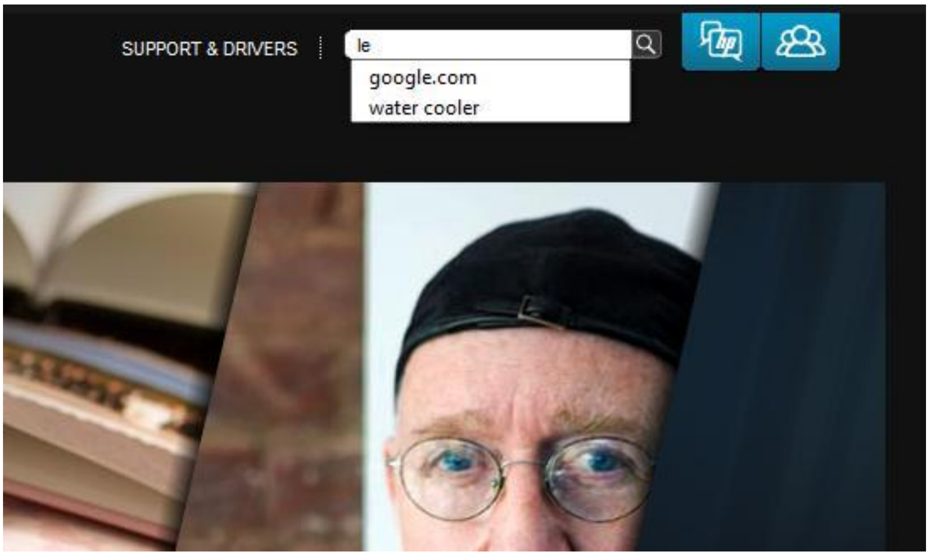
PC+port%E1til+de+entrete...



SUPPORT & DRIVERS

w

- water cooler
- wc
- wiki

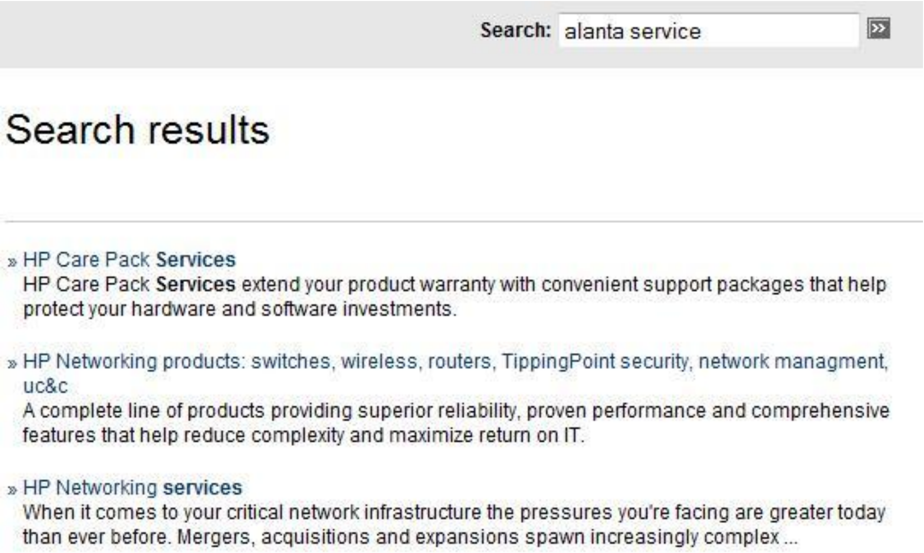


SUPPORT & DRIVERS

le

- google.com
- water cooler

8.2 B



Search: alanta service

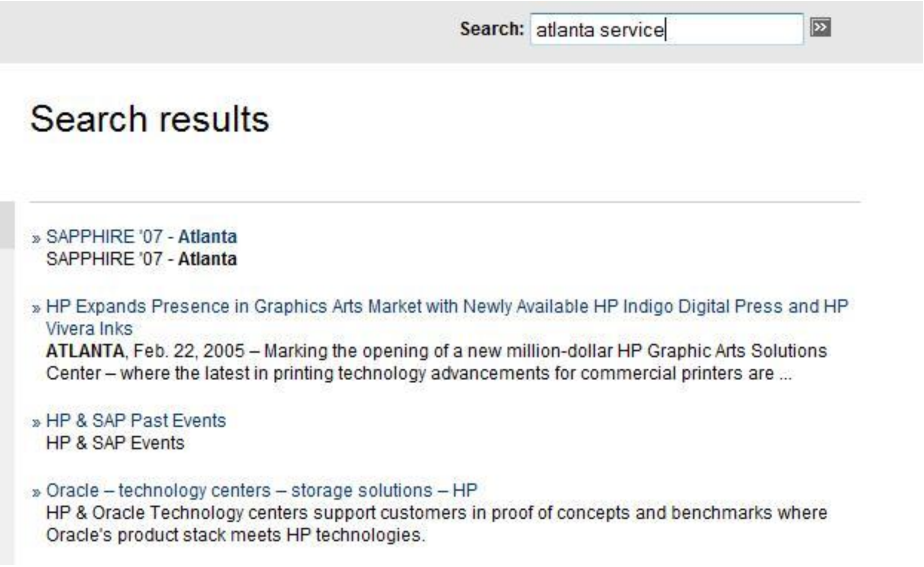
Search results

» HP Care Pack **Services**
HP Care Pack **Services** extend your product warranty with convenient support packages that help protect your hardware and software investments.

» HP Networking products: switches, wireless, routers, TippingPoint security, network managment, uc&c
A complete line of products providing superior reliability, proven performance and comprehensive features that help reduce complexity and maximize return on IT.

» HP Networking **services**
When it comes to your critical network infrastructure the pressures you're facing are greater today than ever before. Mergers, acquisitions and expansions spawn increasingly complex ...

Figure 5. Misspelled query “Alanta service”



Search: atlanta service

Search results

» SAPPHIRE '07 - **Atlanta**
SAPPHIRE '07 - **Atlanta**

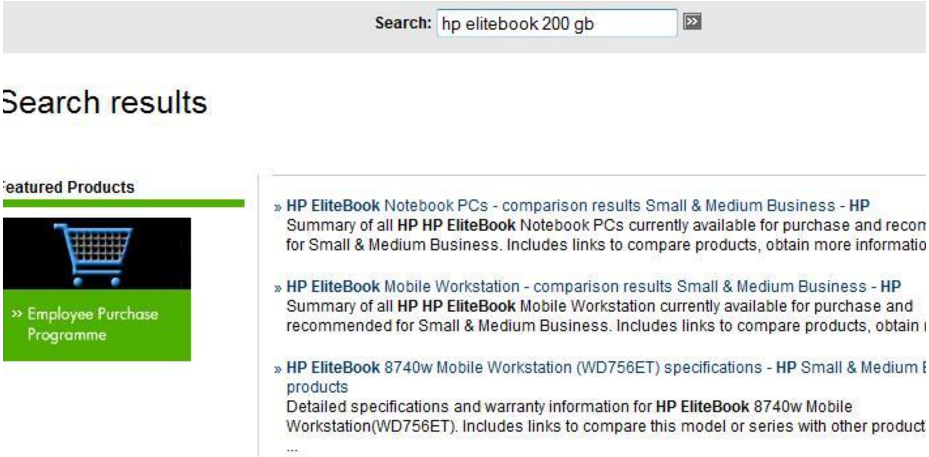
» HP Expands Presence in Graphics Arts Market with Newly Available HP Indigo Digital Press and HP Vivera Inks
ATLANTA, Feb. 22, 2005 – Marking the opening of a new million-dollar HP Graphic Arts Solutions Center – where the latest in printing technology advancements for commercial printers are ...

» HP & SAP Past Events
HP & SAP Events

» Oracle – technology centers – storage solutions – HP
HP & Oracle Technology centers support customers in proof of concepts and benchmarks where Oracle's product stack meets HP technologies.

Figure 6. Search page for query “Atlanta service”

8.3 C



Search: hp elitebook 200 gb

Search results

eatured Products

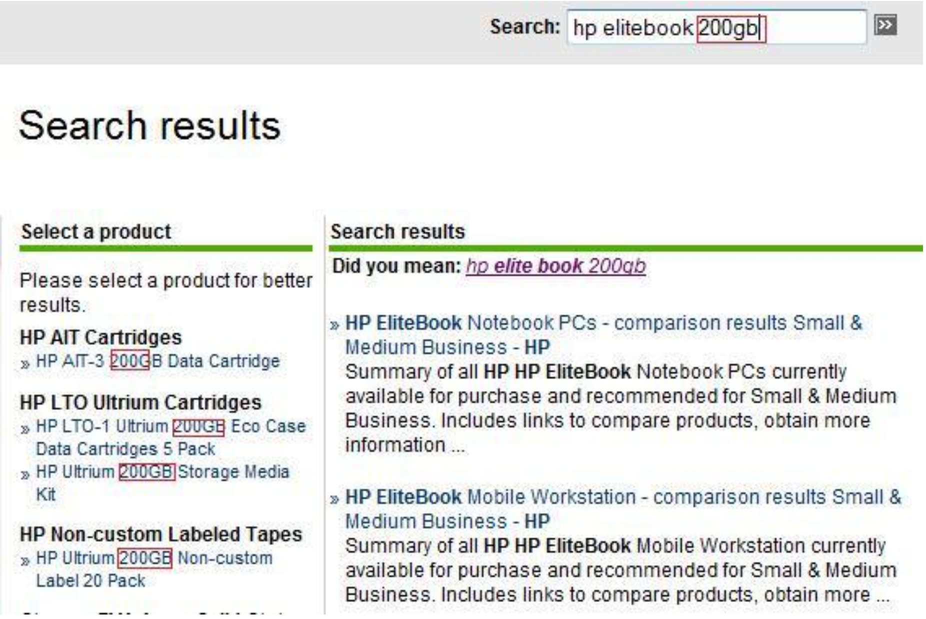
» Employee Purchase Programme

» HP EliteBook Notebook PCs - comparison results Small & Medium Business - HP
Summary of all HP HP EliteBook Notebook PCs currently available for purchase and recon
for Small & Medium Business. Includes links to compare products, obtain more informatio

» HP EliteBook Mobile Workstation - comparison results Small & Medium Business - HP
Summary of all HP HP EliteBook Mobile Workstation currently available for purchase and
recommended for Small & Medium Business. Includes links to compare products, obtain

» HP EliteBook 8740w Mobile Workstation (WD756ET) specifications - HP Small & Medium
products
Detailed specifications and warranty information for HP EliteBook 8740w Mobile
Workstation(WD756ET). Includes links to compare this model or series with other product
...

Figure 7. Search page for query “hp elitebook 200 gb”.



Search: hp elitebook 200gb

Search results

Select a product

Please select a product for better results.

HP AIT Cartridges
» HP AIT-3 200GB Data Cartridge

HP LTO Ultrium Cartridges
» HP LTO-1 Ultrium 200GB Eco Case Data Cartridges 5 Pack
» HP Ultrium 200GB Storage Media Kit

HP Non-custom Labeled Tapes
» HP Ultrium 200GB Non-custom Label 20 Pack

Search results

Did you mean: *hp elite book 200gb*

» HP EliteBook Notebook PCs - comparison results Small & Medium Business - HP
Summary of all HP HP EliteBook Notebook PCs currently available for purchase and recommended for Small & Medium Business. Includes links to compare products, obtain more information ...

» HP EliteBook Mobile Workstation - comparison results Small & Medium Business - HP
Summary of all HP HP EliteBook Mobile Workstation currently available for purchase and recommended for Small & Medium Business. Includes links to compare products, obtain more ...

Figure 8. Search page for query “hp elitebook 200gb”.