Annals of the Academy of Romanian Scientists
Series on Science and Technology of Information
ISSN 2066 - 2742 Volume **7**, Number **1**/**2014**

AUTOMATIC THREE DIMENSION (3-D) WORD ALIGNMENT APPROACH

Abdel Alnasser ALASFOUR[1], Ştefan TRAUŞAN-MATU[2]

Abstract. *A massive effort is needed to build a parallel aligned corpus, so building a tool to for automatic alignment will be useful for natural language processing in general and information retrieval in particular. In our paper we present a new approach which mixed most of the known alignment techniques to achieve high precision and accuracy ratio without human intervention. A list of most English words was used as anchor list following the Pareto principle.*

Keywords: Alignment, Bi-text, Named Entity, Oracle Text

1. Introduction

Parallel corpora are now one of the most important key resources for multilingual natural language processing including machine learning, information retrieval, and machine translation systems [2]. There are many large scale corpora available offline and online on the WEB. Our concern was to find and build a suitable framework for developing an alignment tool to build any parallel aligned corpus in general and building an Arabic-English parallel corpus in particular. The framework we created is using the available functions and procedures of the "Oracle Text" [1].

Our algorithms were developed in order to be applied directly to any target corpus which will be located in database tables. It gives us the ability to manipulate, analyze and evaluate the results for more accuracy. In order to build such a tool we started by investigating the latest methodologies and approaches in the field of bi-text alignment technologies. In the next sections we will describe in further details each step for achieving our main purpose. We start by teaching our system with the most English used words, keeping in our mind the Pareto principle [14], also known as Pareto law's which says "For many events, roughly 80% of the effects come from 20% of the causes".

Therefore, a list of 1000 common English words was translated to Arabic to be as an initial seed for our bilingual dictionary. This was very useful for developing our alignment tool so that we can align any parallel corpus in the next future.

[1]Ph.D. student, Faculty of Automatic Control and Computers, University Politehnica of Bucharest, Bucharest, Romania, (Nasser_Asfour@yahoo.com).
[2]Member of AOSR. Prof., Ph.D., Faculty of Automatic Control and Computers, University Politehnica of Bucharest; Senior Researcher, Research Institute for Artificial Intelligence of the Romanian Academy, Romania, (stefan.trausan@cs.pub.ro).

These techniques can be used to align any other parallel corpus by creating a list of the most used words in both languages in order to facilitate the creating and building an alignment links for those desired parallel corpus. Building a parallel corpus at words' level need a massive implementation efforts for achieving the desired results; starting by finding suitable well translated text files, segmentation, tokenization, stemming, sentences alignment, phrase alignment, words alignment, mapping between the two texts, and finally creating the parallel alignment corpus. In the next sections we will talk about parallel corpus in general. In the "OraLign" section we will describe the methodology we followed to align text.

2. **Related Work**

The main idea of a parallel corpus is a text in language "A" placed alongside with its translation in any other language "B", that means collecting and setup as much parallel text in one huge file known as parallel corpus [2]**.** This huge "parallel corpus" file must satisfy and be applicable in the linguistic domain research such as information retrieval, machine translation and many other applications in the field of natural language processing [6, 7]. The most important process in building a parallel corpus is the "alignment", which is the mapping between the opposite text at many levels, paragraph, sentences and words level. There are many techniques for bilingual corpora alignment [6]. These methods can be categorized in three main categories:

- **Statistical approaches.**

In the statistical approaches there are two major applications that have been introduced. Both of them are length-based approaches, such as the length-based approach by Brown and Lai, which count the words in each sentence before building any alignment link [6]. The approach suggested by Gale and Church also depends on the count of characters in both opposite sentences before creating any alignment link [7, 8].

- **Lexical approaches**

Most of the alignment techniques in this type of alignment depend on lexical sources such as bilingual dictionaries, grammar rule-based.

- **Hybrid approaches**

A combination of statistical and lexical approaches can be used to achieve bilingual corpus alignment

3. **OraLign**

Most of the alignment approaches have been applied to many bilingual corpora and they have been evaluated and have been successful in many applications. Our

main concern was to build an alignment tool "OraLign" for aligning Arabic-English bi-text. With respect to Arabic language the length-based approaches is not the optimal choice due to:

1- Arabic structure of text.
2- Arabic characters type.
3- Grammatical differences between Arabic and English.
4- Arabic rhetoric and syntax.

These differences lead some times to get one into many sentence alignment gaps, or to blank alignment problems. On the other hand, depending only on lexical approaches will not give us the expected results due to many difficulties such as finding a suitable bilingual dictionary. In order to create the OraLign tool we applied a new technique which mixed many of known alignment techniques with extra addition and more modifications.

OraLign as it will be describes in the next sections will be a language-independent word alignment tool. OraLign will mainly depend on an initial bilingual dictionary as a lexical anchor and a new statistical approach called 3-dimension techniques. See Figure 1, which represents OraLign procedure and Figure 2, which shows OraLign three dimensions word alignment approach, where token_text is the word or token in the documents/sentence, token_first contains the ID number of the first document/sentence where that token occurs, while token_last is the ID number of the last document/sentence where that token appears, and finally the token_count will carry out how many times that token appears in all the documents/sentence of the corpus.

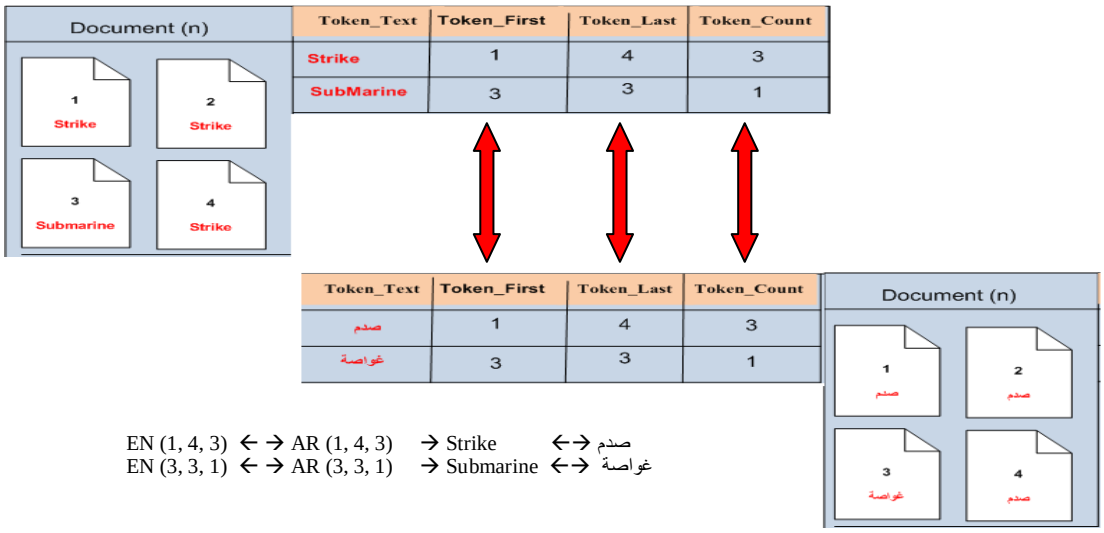

Fig. 1. OraLign Main procedure description with an example.

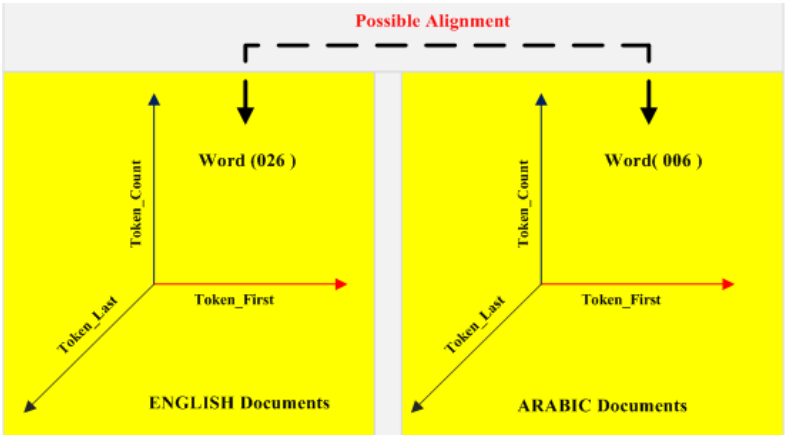

Fig. 2. OraLign 3-dimension approach.

4. Oracle text

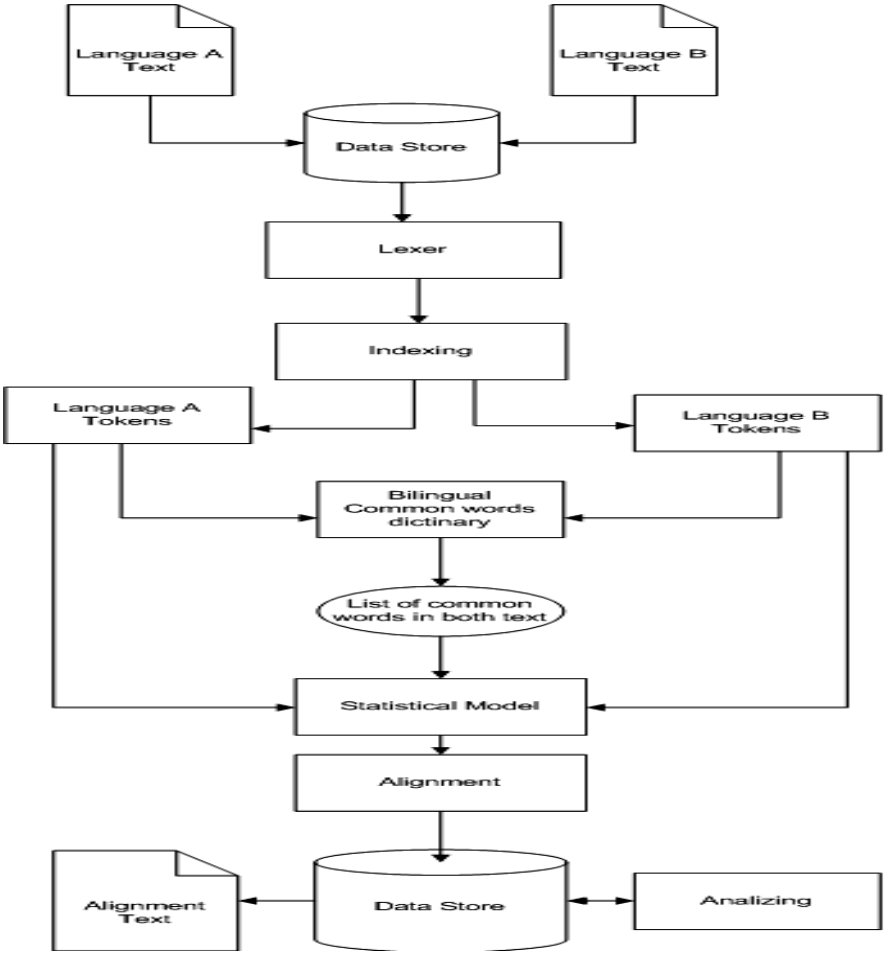

Fig. 3. OraLign implantation process.

Oracle was our first option as a basis for our framework; especially "Oracle Text" which offers a complete text search solution [1]. Another tools come also from Oracle: Developer 6i was used to create the GUI for OraLign tool, both of them running in the Windows OS environment. Our implementation can easily be applied in many different environments (MS Access, MYSQL). Figure 3 shows the implementation process and all the sub-processes, which will be describes in more details in the next sections.

5. OraLign framework

For the purpose of building our framework we decided to include lexical information as anchor points, which contain a list of the most common used words in English and then we translated these words to Arabic using a dictionary. This small dictionary will be the anchor list to establish our alignment algorithm "Initial dictionary". In the next sub-sections we will describe in more details OraLign.

A. Loading documents

Since Oracle supports the processing of any kind of documents (PDF, DOC, Text, etc.) with a massive support for most of different languages, the user needs to setup and configure the appropriate database tables to keep and save these documents.

B. Lexer

The main purpose for the lexer step is to split the documents into tokens according to the specified language of the document and the setting of the configuration parameters for that language; which include the declaration of the sentence borders (‘.’, ’?’, ’!’), whitespace (' ') as a separation character between document words, or any specific characteristics setting [1]. The output of this process is raw data of document tokens. Figure 4, shows a sample of document tokens after lexer processing.

{ REPORT} , { USA } , { RESULT } , { STATEMENT } , { DAMAGE } , { INJURY } , { OCCURRENCE } , { ESTABLISH } , { FIVE } , { WITHOUT } , { MAJOR } , { COLLIDE } , { GULF } , { PERISCOPE } , { SUBMARINE } , { FLEET } , { SHIP } , { IDENTITY }

Fig.4. English document token list.

C. Indexing

There are many types of indexes that Oracle can support. For our purpose we implemented CONTEXT as an index option to maximize the ability of searching and locating any token no matter how large are the documents. Since the documents are stored in the database tables, it was very easy to select the most appropriate index option. Context indexing process creates several auxiliary tables [1]. One of these tables is what is known as the "I” or "Token List" table which contains all the document tokens as rows and it has many useful attributes.

The token list table "I" also contains information for linking the tokens to their document source. Context index supports most of known languages especially English language and it also supports Arabic with some attention and with suitable configurations. Figure 5, represents a sample of OraLign token list table and its main attributes which are:

1- Token text.
2- Token first: the ID number of the sentence /document in which the token appears for the first time.
3- Token last: the ID number of the sentence/document in which the token appears for the last time.
4- Token count: how many times that token appears in the document(s).

	TOKEN_TEXT	TOKEN_FIRST	TOKEN_LAST	TOKEN_COUNT	WORD_ORDER
1	IDENTITY	1	1	1	020
2	ESTABLISH	1	1	1	023
3	WITHOUT	1	4	2	024
4	RESULT	1	1	1	025
5	DAMAGE	1	5	3	028
6	OCCURRENCE	1	1	1	031

Fig. 5. Modified token list with it is main attributes.

D. Bilingual common words dictionary

This dictionary contains 1000 of the most common used English words. It was collected and translated to Arabic in a direct way. We used this list to train our algorithm. Figure 6, shows a sample list taken from the initial bilingual dictionary used in our framework.

	ENGLISH	ARABIC
1	WATER	الماء
2	THAN	من
3	CALL	النداء
4	FIRST	أولا
5	WHO	من
6	MAY	مايو

Fig.6. Sample of "1000" startup dictionary.

In this step a reference table creates a mapping between tokens in the startup dictionary table and the token list table for each token that appears in both lists. In other words, if any of the documents tokens is found in the startup dictionary a reference link will be created and saved in a table.

E. OraLign Statistical model

Depending on the output of each process, a statistical model is initialized to analyze each token's property and check if there is any ambiguity before building and creating a possible alignment link [11, 12].

Figure 7 shows a situation where two tokens have the same values for token first, token last and token count and it seems to be a possible alignment link that can be created between them. So, before building and creating this link, the statistical process will check all the tokens in both texts for any tokens which have the same attributes values. If the model finds any other tokens having the same attribute values then it will check the startup dictionary for the meaning of the tokens in both languages. If it exists, then it will check the dictionary values for both tokens. If it is the same then a link will be created, if no then the system will perform a second cycle after removing all the tokens that have been already linked.

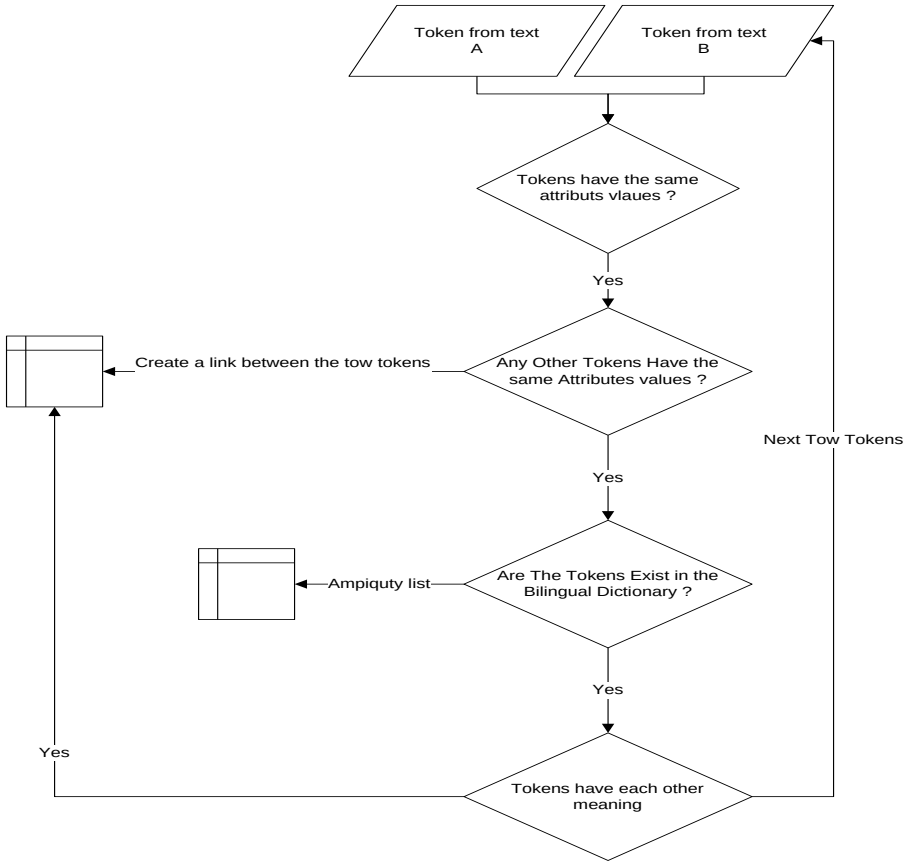

Fig. 7. English document token list.

F. Alignment Process

After all the previous steps have taken place, the alignment process will start as shown in the alignment procedure. Figure 8, represents two parallel texts in English and Arabic has been loaded to OraLign tables. Figures 9 and 10 represent the tokens list for both documents after they have been loaded and indexed. Therefore, the alignment process which include several sub-steps, starts by

checking if there is any token in the startup dictionary, name entity, and any similar words exist in both documents/sentences [11]. In the next sections we will describe each sub-process in detail.

English	Arabic
The periscope of a US submarine collided in the Gulf the day before yesterday, Thursday, with a ship whose identity was not established, Without resulting in major damage, or the occurrence of injuries, as the USA Fifth Fleet reported in a statement. The statement advised that the submarine Jacksonville, which is of the class Los Angeles, struck a ship during an operation in the Gulf, at 5.00 in the morning of Thursday, local time (2.00 GMT). The statement added that the submarine surfaced following the incident, to ascertain whether the ship , which was not identified, had received damage or not . But the ship continued moving on the same course and at the same speed, without sending out a distress call. One of the two periscopes of the submarine was damaged, but the incident did not affect its nuclear reactor and propulsion engine.	صدم منظار غواصة أميركية في الخليج ، أول من أمس الخميس، سفينة لم تحدد هويتها ، من دون التسبب في أضرار كبيرة أو حدوث إصابات . كما أفاد الأسطول الأميركي الخامس في بيان . وهي من نوع لوس أنجليس ، جاكسونفيل وأفاد البيان بأن الغواصة صدمت سفينة خلال عملية في الخليج في الخامسة صباح الخميس بالتوقيت المحلي الثانية بتوقيت غرينيتش . وأضاف البيان أن الغواصة طفت إثر الحادث ، للتحقق مما إذا كانت السفينة التي لم يتم التعرف عليها أصيبت بأضرار أم لا . لكن السفينة واصلت سيرها في الوجهة نفسها وبالسرعة نفسها من دون إطلاق نداء استغاثة ، و تضرر أحد مناظري الغواصة و لم يؤثر الحادث على مفاعلها النووي ومحرك الدفع .

Fig. 8. Two parallel texts in English and Arabic.

	TOKEN_TEXT	TF	TL	TC	WO1
1	PERISCOPE	1	5	2	002
2	USA	1	1	2	005
3	SUBMARINE	1	5	4	006
4	COLLIDE	1	1	1	007
5	GULF	1	2	2	010
6	SHIP	1	4	4	018
7	IDENTITY	1	1	1	020
8	ESTABLISH	1	1	1	023
9	WITHOUT	1	4	2	024
10	RESULT	1	1	1	025
11	MAJOR	1	1	1	027
12	DAMAGE	1	5	3	028
13	OCCURRENCE	1	1	1	031
14	INJURY	1	1	1	033
15	FIVE	1	2	2	037
16	FLEET	1	2	2	038
17	REPORT	1	1	1	039
18	STATEMENT	1	3	3	042
19	ADVISE	2	2	1	003
20	JACKSONVILLE	2	2	1	007
21	CLASS	2	2	1	012
22	LOS	2	2	1	013
23	ANGELES	2	2	1	014
24	STRIKE	2	2	1	015
25	DURING	2	2	1	018
26	OPERATION	2	2	1	020
27	MORNING	2	2	1	028
28	LOCAL	2	2	1	031
29	TIME	2	2	1	032
30	GREENWICH	2	2	1	034
31	ADD	3	3	1	003
32	SURFACE	3	3	1	007
33	FOLLOWING	3	3	1	008
34	INCIDENT	3	5	2	010
35	ASCERTAIN	3	3	1	012
36	IDENTIFY	3	3	1	019
37	RECEIVE	3	3	1	021
38	CONTINUE	4	4	1	004
39	MOVING	4	4	1	005
40	SAME	4	4	4	008
41	COURSE	4	4	1	009
42	SPEED	4	4	1	014
43	SEND	4	4	1	016
44	OUT	4	4	1	017
45	DISTRESS	4	4	1	019
46	CALL	4	4	1	020
47	AFFECT	5	5	1	016
48	NUCLEAR	5	5	1	018
49	REACTOR	5	5	1	019
50	PROPULSION	5	5	1	021
51	ENGINE	5	5	1	022

Fig. 9. English document token list.

	TOKEN_TEXT	TF	TL	TC	WO1
1	صدم	1	2	2	001
2	منظار	1	5	2	002
3	غواصة	1	5	4	003
4	أميركي	1	1	2	004
5	خليج	1	2	2	006
6	سفينة	1	4	4	011
7	حدد	1	1	1	013
8	هويه	1	1	1	014
9	دون	1	4	2	016
10	تسبب	1	1	1	017
11	ضرر	1	5	3	019
12	كبير	1	1	1	020
13	حدوث	1	1	1	022
14	إصابة	1	1	1	023
15	أفاد	1	2	2	025
16	أسطول	1	1	1	026
17	خامس	1	2	2	028
18	بيان	1	3	3	030
19	خلال	2	2	1	008
20	عملي	2	2	1	009
21	صباح	2	2	1	014
22	وقت	2	2	1	016
23	محلي	2	2	1	017
24	غرينيتش	2	2	1	020
25	نوع	2	2	1	024
26	لوس	2	2	1	025
27	أنجليس	2	2	1	026
28	جاكسونفيل	2	2	1	027
29	أضاف	3	3	1	002
30	طفا	3	3	1	006
31	إثر	3	3	1	007
32	حادث	3	5	2	008
33	تحقق	3	3	1	009
34	تم	3	3	1	016
35	تعرف	3	3	1	017
36	أصيب	3	3	1	019
37	واصل	4	4	1	003
38	سير	4	4	1	004
39	وجهة	4	4	1	006
40	نفس	4	4	2	007
41	سرعة	4	4	1	009
42	إطلاق	4	4	1	013
43	نداء	4	4	1	014
44	استغاثة	4	4	1	015
45	أثر	5	5	1	008
46	مفاعل	5	5	1	011
47	نووي	5	5	1	012
48	محرك	5	5	1	014
49	دفعة	5	5	1	015

Fig. 10. Arabic document token list.

1- Start-up dictionary:

Figure 11 shows the tokens that have been translated using the startup dictionary. For more accuracy the system must be sure that both sides of the dictionary tokens exist in the opposite document to avoid any miss-translation errors. In other words, if the English word is founded in the dictionary and the translated word does not exist in the Arabic document; it will be neglected. In our example OraLign found 12 tokens and they are ready to be linked to each other.

	EN_TOKEN	EN_TF	EN_W_ORDER	AR_TOKEN	AR_TF	AR_W_ORDER
1	SHIP	1	018	سفينة	1	011
2	MAJOR	1	027	كبير	1	020
3	CLASS	2	012	نوع	2	024
4	DURING	2	018	خلال	2	008
5	MORNING	2	028	صباح	2	014
6	TIME	2	032	وقت	2	016
7	ADD	3	003	أضاف	3	002
8	CONTINUE	4	004	واصل	4	003
9	SAME	4	008	نفس	4	007
10	SPEED	4	014	سرعة	4	009
11	CALL	4	020	نداء	4	014
12	ENGINE	5	022	محرك	5	014

Fig. 11. List of tokens founded in the dictionary.

2- Named Entity Recognition

In many cases the Arabic document contains named entities for persons and places [4]. When these names are translated from English to Arabic or vice versa, they will be written using the target language characters and depends on the source language pronunciation for that name; as an example, the country name "Romania" will be written in Arabic as "رومانيا" which is the same pronunciation as it is in English language. For that reason we create a special procedure to extract the named entities from Arabic document and then we compare them with those in the English document [13, 14]. Figure 12, shows the named entities which have been founded in both documents.

	TOKEN_TEXT	EN_TF	EN_TL	EN_TC	EN_W	AR_TOKEN	AR_TF	AR_TL	AR_TC	AR_W	JWS
1	JACKSONVILLE	2	2	1	007	جاكسونفيل	2	2	1	027	88
2	LOS	2	2	1	013	لوس	2	2	1	025	91
3	ANGELES	2	2	1	014	أنجليس	2	2	1	026	82
4	GREENWICH	2	2	1	034	غرينيتش	2	2	1	020	82

Fig. 12. A list of the names entity.

3- Similar tokens extraction

In this step, OraLign will locate and extract any similar tokens found in both documents. Many of Arabic documents that we considered are a mixture of scientific or medical articles. In such documents you will find foreign words mainly in Latin, which are written as same as they are in the original documents. As an example, Figure 13 shows two similar tokens that appear in bi-text.

English	Arabic
Mitsubishi introduces 2002 Pajero new features! In August, Mitsubishi announced the new standard and optional equipment on the 2002 Pajero. New standard equipment includes INVECS-II 4 automatic transmission with Sports Mode and discharge headlamps, while front seatback pockets and illuminated vanity mirrors with lids fitted on both sun visors are among the standard utility features.	ميتسوبيتشي تطرح طرازها الجديد من باجيرو ٢٠٠٢ بمميزات جديدة ! قامت ميتسوبيتشي في شهر أغسطس ، بالإعلان عن التجهيزات الاختيارية و القياسية الجديدة التي تهيئت بها باجيرو ٢٠٠٢ . حيث تضمنت التجهيزات الجديدة ناقل حركة أوتوماتيكي من الجيل الثاني INVECS-II 4 بأسلوب تشغيل رياضي ، كشافات أمامية مفرغة ، في حين تم تزويد خلفية المقاعد الأمامية بجيوب لحفظ الأوراق و الملفات مع مرايا أنيقة مضيئة مزودة بأغطية متينة على واقيات الشمس للسائق و مرافقه و هي تجهيزات و مميزات جديدة قياسية .

Fig.13. Similar tokens example.

After the three previous steps finished, OraLign will remove out all the tokens from both token list tables, and the remaining tokens will be moved forward to the main procedure of OraLign which is the 3-D approach. In the 3-D procedure there will be as many cycles as are needed to align as much as possible tokens in both documents. For that reason the token list will be divided depending on the token_first value. So all the tokens which are in the first document/sentence will be in one group (sub-list) with TF=1, and so on. The tokens in the sub-list will be sorted in descending order based on the value of word order column. This step is prerequisite for OraLign to start searching and mapping any possible alignment tokens. Figure 14 presents an example of how the tokens list is divided to many sub-list depending on how many documents are there in the corpus.

	TOKEN_TEXT	W	TC	TL	TF
1	REPORT	039	1	1	1
2	FLEET	038	1	1	1
3	INJURY	033	1	1	1
4	OCCURRENCE	031	1	1	1
5	RESULT	025	1	1	1
6	ESTABLISH	023	1	1	1
7	IDENTITY	020	1	1	1
8	COLLIDE	007	1	1	1
9	USA	005	2	1	1
10	FIVE	037	2	2	1
11	GULF	010	2	2	1
12	STATEMENT	042	3	3	1
13	WITHOUT	024	2	4	1
14	PERISCOPE	002	2	5	1
15	DAMAGE	028	3	5	1
16	SUBMARINE	006	4	5	1

	TOKEN_TEXT	TF	TL	TC	WO1
1	ADVISE	2	2	1	003
2	STRIKE	2	2	1	015
3	OPERATION	2	2	1	020
4	LOCAL	2	2	1	031

	TOKEN_TEXT	TF	TL	TC	WO1
1	SURFACE	3	3	1	007
2	FOLLOWING	3	3	1	008
3	INCIDENT	3	5	2	010
4	ASCERTAIN	3	3	1	012
5	IDENTIFY	3	3	1	019
6	RECEIVE	3	3	1	021

	TOKEN_TEXT	TF	TL	TC	WO1
1	MOVING	4	4	1	005
2	COURSE	4	4	1	009
3	SEND	4	4	1	016
4	OUT	4	4	1	017
5	DISTRESS	4	4	1	019

Fig.14. List of tokens in sub-list TF=1 ,TF=2 ,TF=3 and TF=4

6. **Practical alignment process**

After collecting all the information, the system is now ready to begin and build any possible alignment link between the appropriate suitable tokens from both documents. To develop our algorithms we applied our new method "3-D" alignment approach. First, the system removes out all the translation tokens, named entities, and similar tokens from both token lists and keeps all the other tokens which need to be mapped and link [10]. In our example, the final remaining tokens that need to be aligned are 50 tokens after removing 12 translated tokens and 4 tokens have been linked as named entity "none tokens are in the similar list". In the next section we demonstrate an alignment process for the remaining tokens in sentence one (TF=1) as an example of how our algorithm will work.

Sentence (TF=1) analyzing and mapping

STEP 1

	TOKEN_TEXT	W	TC	TL	TF
1	STATEMENT	042	3	3	1
2	REPORT	039	1	1	1
3	FLEET	038	1	1	1
4	FIVE	037	2	2	1
5	INJURY	033	1	1	1
6	OCCURRENCE	031	1	1	1
7	DAMAGE	028	3	5	1
8	RESULT	025	1	1	1
9	WITHOUT	024	2	4	1
10	ESTABLISH	023	1	1	1
11	IDENTITY	020	1	1	1
12	GULF	010	2	2	1
13	COLLIDE	007	1	1	1
14	SUBMARINE	006	4	5	1
15	USA	005	2	1	1
16	PERISCOPE	002	2	5	1

	TF	TL	TC	W	TOKEN_TEXT
1	1	3	3	030	بيان
2	1	2	2	028	خامس
3	1	1	1	026	أسطول
4	1	2	2	025	أ فاد
5	1	1	1	023	إصابة
6	1	1	1	022	حدوث
7	1	5	3	019	ضرر
8	1	1	1	017	تسبب
9	1	4	2	016	دون
10	1	1	1	014	هويه
11	1	1	1	013	حدد
12	1	2	2	006	خليج
13	1	1	2	004	أميركي
14	1	5	4	003	غواصة
15	1	5	2	002	منظار
16	1	2	2	001	صدم

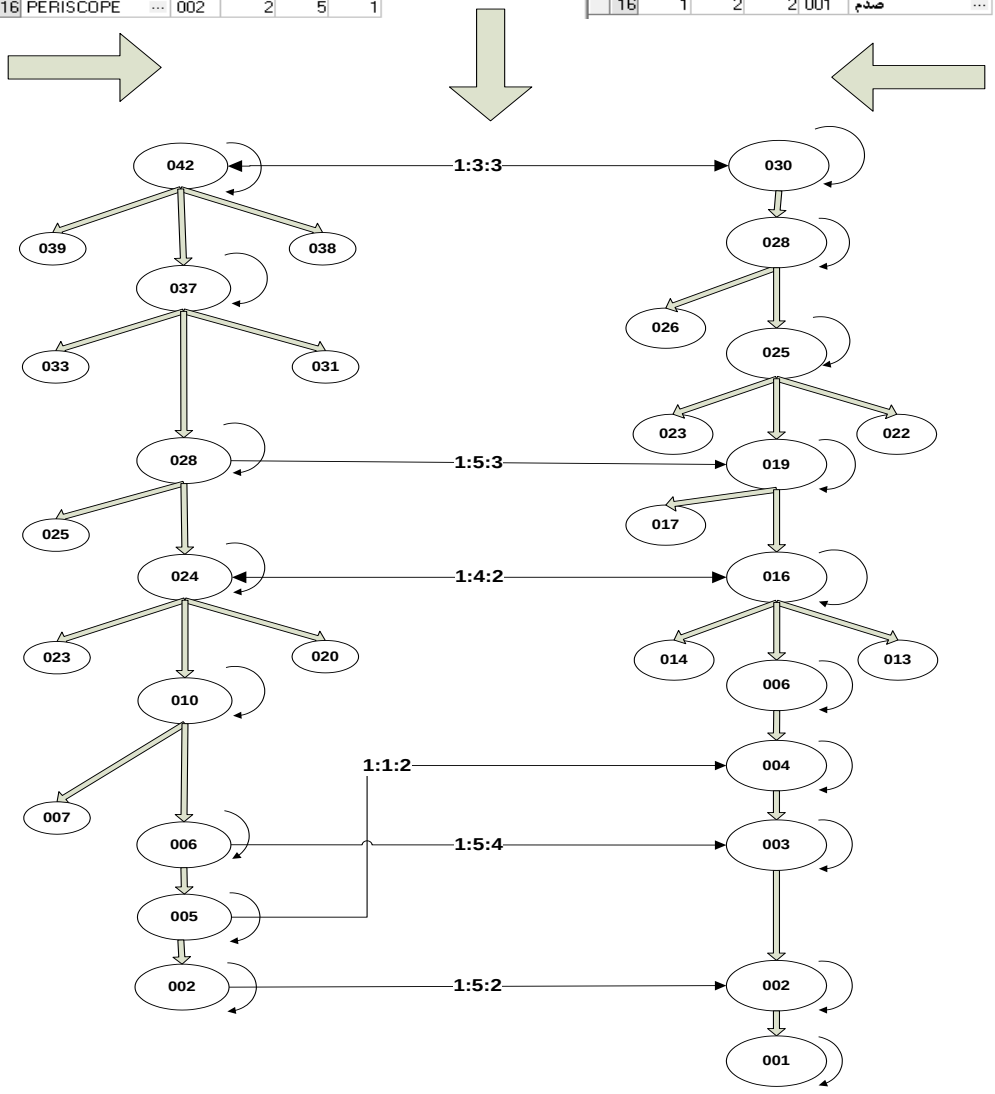

Fig. 15. Tokens tree for TF=1.

In this step the system builds a tree in which each token is represented as a circle, each token with more than one occurrence is presented as a circle with recursive arrow, and each token with occurrence 1 is presented as a simple circle. Any circle with arrow –occurrence more than one- can be linked to any other circle as father relation, but the simple circle will not be able to connect to any other token in the same list.

STEP 2

After removing out the matching tokens from step (1) the system will divide the tokens in TF=1 to several parts. Each part border will be the removing tokens - see Figure 16.

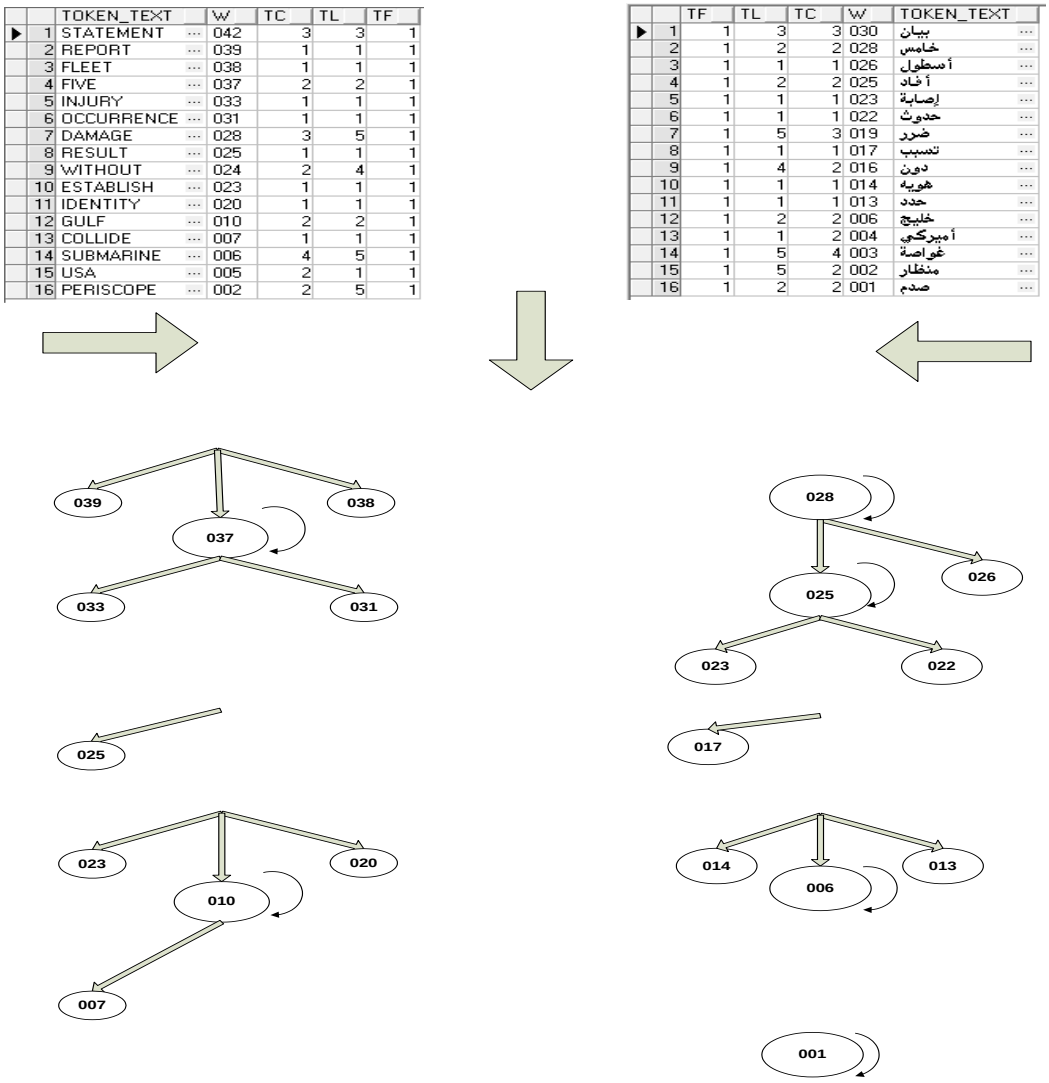

Fig. 16. List of tokens in sentence "1" – TF=1.

Before building such a tree the system will sort the tokens in descending order depending on the token order in the sentence. Then OraLign will start to build the first token which has the high order (last word) in the sentence. Figure 15, shows the tree and the arrows and also shows the way it is be created. For English text the tree and its branches are created from left to right as it is the same in reading the English text. On the other side the tree for Arabic document will be created from right to left as it is the same when reading Arabic text. After setting up the tree for both documents, the system will compare the tokens attributes values from up to down depending on the value of TF, TL, TC and then build a link between those tokens. If there is any suspicion of ambiguity in the tree caused by many tokens having the same values for TF, TL, and TC, the system will pass to the next token. When the first cycle is finished the system will divide the sub list into many extra lists after removing out the tokens which already have been linked to each other in the first rotate.

	TOKEN_TEXT	W	TC	TL	TF
1	REPORT	039	1	1	1
2	FLEET	038	1	1	1
3	FIVE	037	2	2	1
4	INJURY	033	1	1	1
5	OCCURRENCE	031	1	1	1

	TF	TL	TC	W	TOKEN_TEXT
1	1	2	2	028	خامس
2	1	1	1	026	أسطول
3	1	2	2	025	أخاد
4	1	1	1	023	إصابة
5	1	1	1	022	حدوث

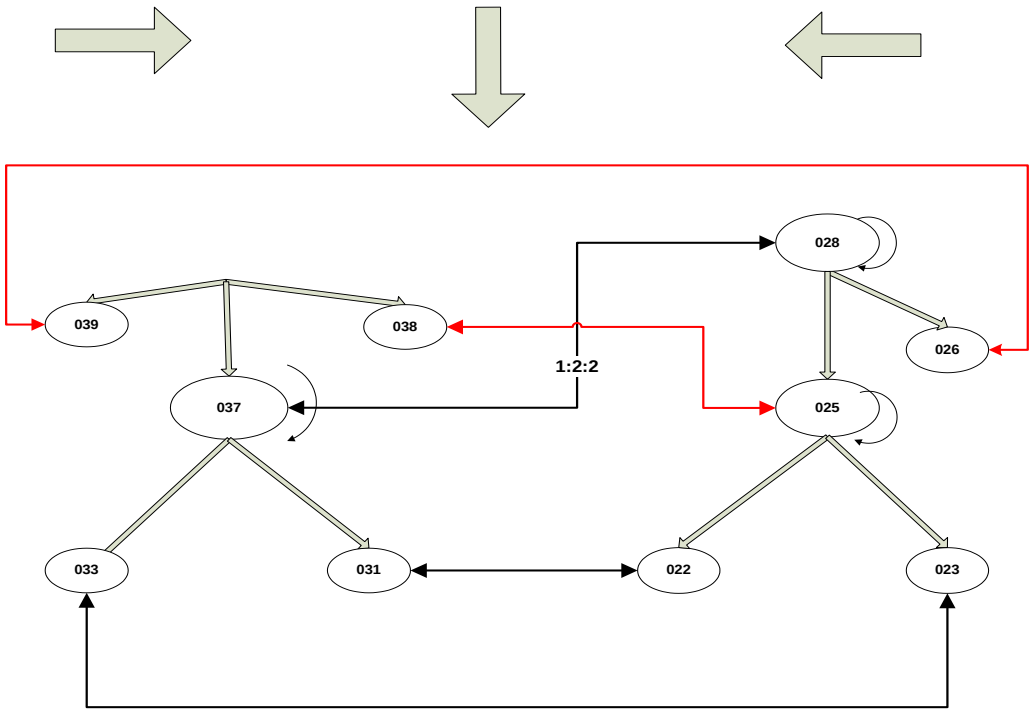

Fig. 17. Tokens tree for TF=1 , Part=1.

STEP 3

The first divided part of TF=1, will start mapping from up to down and from the right side of Arabic tokens to the left side of English tokens as shown in figure 17.

STEP 4

In this step a direct link will be build since there is just one token in each side, see figure 18.

	TF	TL	TC	W	TOKEN_TEXT
1	1	1	1	017	تسبب

	TOKEN_TEXT	W	TC	TL	TF
1	RESULT	025	1	1	1

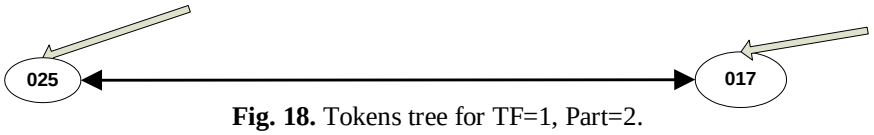

Fig. 18. Tokens tree for TF=1, Part=2.

STEP 5

In the final step for TF=1 as shown in Figure 19, there exist an Arabic token which has occurrence value more than one and that token have been linked to English token with occurrence value equal to one. In this case the system will keep in mind –memory - that Arabic token and move a copy of it to the next sentence to try to find any dominated English token in that sentence .

	TOKEN_TEXT	W	TC	TL	TF
1	ESTABLISH	023	1	1	1
2	IDENTITY	020	1	1	1
3	GULF	010	2	2	1
4	COLLIDE	007	1	1	1

	TF	TL	TC	W	TOKEN_TEXT
1	1	1	1	014	هويه
2	1	1	1	013	حدد
3	1	2	2	006	خليج
4	1	2	2	001	صدم

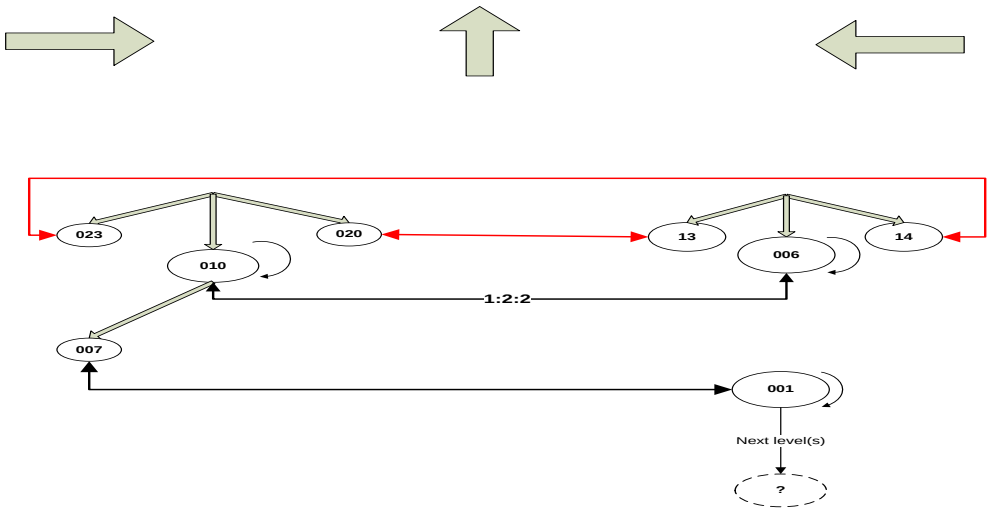

Fig. 19. Tokens tree for TF=1, Part=3.

TF = 1 alignment outcomes

❖ **Step 1 output →**

	EN_TOKEN	EN_W	EN_TC	EN_TL	EN_TF	AR_TF	AR_TL	AR_TC	AR_W	AR_TOKEN
1	PERISCOPE	002	2	5	1	1	5	2	002	منظار
2	USA	005	2	1	1	1	1	2	004	أميركي
3	SUBMARINE	006	4	5	1	1	5	4	003	غواصة
4	WITHOUT	024	2	4	1	1	4	2	016	دون
5	DAMAGE	028	3	5	1	1	5	3	019	ضرر
6	STATEMENT	042	3	3	1	1	3	3	030	بيان

Fig.20. Tokens tree for TF=1, step=1

❖ **Step 3 output**

	EN_TOKEN	EN_W	EN_TC	EN_TL	EN_TF	AR_TF	AR_TL	AR_TC	AR_W	AR_TOKEN
1	OCCURRENCE	031	1	1	1	1	1	1	022	حدوث
2	INJURY	033	1	1	1	1	1	1	023	إصابة
3	FIVE	037	2	2	1	1	2	2	028	خامس
4	FLEET	038	1	1	1	1	2	2	025	أفاد
5	REPORT	039	1	1	1	1	1	1	026	أسطول

Fig.21. Tokens tree for TF=1, step=3

❖ **Step 4 output**

	EN_TOKEN	EN_W	EN_TC	EN_TL	EN_TF	AR_TF	AR_TL	AR_TC	AR_W	AR_TOKEN
1	RESULT	025	1	1	1	1	1	1	017	تسبب

Fig.22. Tokens tree for TF=1, step=4

❖ **Step 5 output**

	EN_TOKEN	EN_W	EN_TC	EN_TL	EN_TF	AR_TF	AR_TL	AR_TC	AR_W	AR_TOKEN
1	COLLIDE	007	1	1	1	1	2	2	001	صدم
2	GULF	010	2	2	1	1	2	2	006	خليج
3	IDENTITY	020	1	1	1	1	1	1	013	حدد
4	ESTABLISH	023	1	1	1	1	1	1	014	هويه

Fig.23. Tokens tree for TF=1, step=5

7. The analysis of the OraLign results

Table 1 and Table 2 present the details about both English and Arabic documents respectively. The percentage share of the bilingual dictionary in the alignment process was “24%”. While the percentage shares of named entities extractions process was “8%” that leaves “68%” for the 3-D share in the whole alignment process. OraLign will give more accuracy result when align large number of documents.

Table 1

Sentence ID "TF"	Number Of Tokens	Tokens In Startup Dictionary	Named Entities	Remaining Tokens Count
1	18	2	0	16
2	12	4	4	4
3	7	1	0	6
4	8	4	0	4
5	5	1	0	4
TOTAL	50	12	4	35

Table 2

Sentence ID "TF"	Number Of Tokens	Tokens In Startup Dictionary	Named Entities	Remaining Tokens Count
1	18	2	0	16
2	10	4	4	2
3	7	1	0	6
4	8	4	0	4
5	5	1	0	4
TOTAL	48	12	4	32

8. **OraLign evaluation**

For evaluating our method we used two documents, each one a translated version of the other. Both documents contain 5 sentences and both of them contain a lot of what are called stop words such as "in, on, to, the, this" in the English document and "من, في, على…" in Arabic. Figures 24 and 25 represent a list of stop words that are removed from both documents before running any further steps.

	ID	STOPW	LANG
1	1	IN	EN
2	2	THE	EN
3	3	AND	EN
4	4	ON	EN
5	5	WITH	EN
6	6	A	EN
7	7	OF	EN
8	8	TOO	EN
9	9	OR	EN
10	10	ALSO	EN

Fig. 24. English stop words sample.

	ID	STOPW	LANG
1	1	من	AR
2	2	في	AR
3	3	على	AR
4	4	الى	AR
5	5	و	AR
6	6	عن	AR
7	7	تم	AR
8	8	مع	AR
9	9	هي	AR
10	10	أيضا	AR

Fig.25. Arabic stop words sample

After removing out the stop words from both documents; the number of remaining tokens was 50 for the English document and 48 in the Arabic document. For instance the maximum number of links OraLign can build is 50 alignment relations (English tokens).

- Tokens in the start-up dictionary are 12
- Tokens in the Named Entity list are 4
- Tokens in the OraLign List are 33

The final number of tokens in all ways are (49), and the reason for not reaching the maximum number of possible link is that one of the English tokens has not been linked to any Arabic tokens, which is "ADVISE" see figure 26.

	TOKEN_TEXT	W	TC	TL	TF
1	ADVISE	003	1	2	2

Fig. 26. English tokens not aligned.

On the other hand, there exists one Arabic token that has been linked with two different English words from the English list, which is ('صدم'), and both of them are correct, see figure 27:

	EN_TOKEN	EN_W	EN_TC	EN_TL	EN_TF	AR_TF	AR_TL	AR_TC	AR_W	AR_TOKEN
1	COLLIDE	007	1	1	1	1	2	2	001	صدم
2	STRIKE	015	1	2	2	1	2	2	001	صدم

Fig. 27. Same arabic token linked with two different english tokens.

Depending on the output of all previously steps we can evaluate our algorithms by calculating the Precision, Recall and f-measure (f-score) for checking the accuracy and error rate for our method [3, 4].

Precision and Recall are the most know basic measures to evaluate finding a specific relevant item within a huge list of items [3, 4]. In further details recall is used to "calculate the ratio of the number of relevant records retrieved to the total number of relevant records in the database". In the other hand precision is used to “measure the ratio of number of relevant records retrieved to the total number of irrelevant and relevant records retrieved". Both of "Recall" and "Precision" are usually expressed as a Percentage. Figure 28, describes "Recall" and "Precision” for any information retrieval system in general.

In our case the total number of records (tokens) is 50. The number of tokens that have been linked was 49, and the correct relations were 43. Suppose we present our results in suitable variables such as:

- Number of relevant tokens linked.43
- Number of relevant tokens not linked. 1
- Number of irrelevant tokens retrieved.7

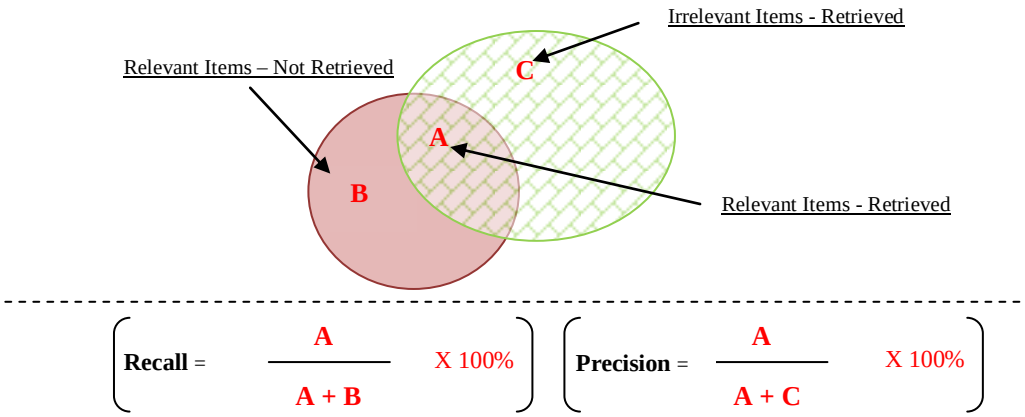

Fig. 28. Recall and Precision descriptions and Formulas.

Since we know that 6 of the relations are not correct we can compute and find out A, B and C:

A=49-6 → 43, B=50-43→ 7, C=49-43→ 6.

From the above values we can calculate and compute both Recall and Precision respectively:

- Recall = A/(A+B) ➔ 43/(43+7) ➔ 86%
- Precision = A/(A+C) ➔ 43/(43+6) ➔ 87%

TF	Recall	Precision	F_measure
1	0.78	0.78	0.78
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	0.88	1.00	0.93
5	0.67	0.67	0.67
	0.86	**0.88**	**0.87**

Fig. 29. Recall and Precision chart.

For more evaluations we can compute the value of f-measure (f-score) which is normally used to measure overall "search" accuracy by depending on the outcomes of both recall and precision [3]. Formula 1and 2, shows the f_measure (f-score) standard formula and it is result.

$$f_{measure} = 2.\frac{Precisin\ .Recall}{Precision+Recall} \quad \text{— 1}$$

$$f_{measure} = 2 * \frac{0.87*0.86}{0.87+0.86} = 0.86 \quad \text{— 2}$$

Figure 29 shows the results of Recall, Precision, and F_measure for each sentence.

9. **Conclusion and future works**

In this paper we have introduced a novel method for bi-text alignment at words level and this is done depending on 1000 common English words which include the stop words.

Next step was the building of a tool for automatic tokens' alignment, which was described and evaluated.

This method can be applied to any bilingual set of files (corpus).

Oracle in general was a perfect option for planning, creating and testing OraLign tool.

Furthermore, Oracle text in particular with its useful utilities gives a massive support for information retrieval.

Since OraLign evaluation results shown an accepted result in terms of Recall, Precision and f_measure [3, 9], in next future we will try to maximize accuracy ratio by train OraLign with different categories of bi-text.

R E F E R E N C E S

[1] Cathy Shea, Oracle Text Reference, 11g Release 2 (11.2). Part No. E24436-04. Oracle Corporation. February **2014**: http://docs.oracle.com/

[2] Johansson, S. **1995**. The approach of the Text Encoding Initiative to the encoding of spoken discourse. In Spoken English on Computer, eds. G. Leech, G. Myers and J. Thomas, 82-98. Harlow: Longman.

[3] William E. Underwood, Matthew G. Underwood , Evaluation of Document Retrieval Technologies to Support Access to Presidential Electronic Records PERPOS Technical Report ITTL/CISTD 02-3,December **2002**. http://perpos.gtri.gatech.edu/

[4] Mohammad Salameh, Improving the Accuracy of English-Arabic Statistical Sentence Alignment ,The International Arab Journal of Information Technology, Vol. 8, No. 2, April **2011**.

[5] Jorg Tiedemann. **1999**. Word alignment - step by step. In Proceedings of the 12th Nordic Conference on Computational Linguistics, pages 216–227, University of Trondheim, Norway.

[6] Brown P. and Lai J., "Aligning Sentences in Parallel Corpora," in the Proceedings of 29th, Annual Meeting for ACL, pp. 169-179, **1991**.

[7] Gale W. and Church K., "A Program for Aligning Sentences in Bilingual Corpus," in the Proceedings of 29th Annual Meeting of the ACL, pp. 177-184, **1991**.

[8] Gale, W. A. and Church, K.W. **1991**a. Identifying word correspondences in parallel texts. In Proc. of the DARPA Workshop on Speech and Natural Language, pages 152–157.

[9] Olson, D. and Delen, D. **2008**. Advanced data mining techniques. 2008 Springer-Verlag Berlin Heidelberg, Library of Congress Control Number: 2007940052. Performance Evaluation for Predictive Modeling 9, 137-140.

[10] Do et al., 2002, Hong-Hai Do, Sergei Melnik, and Erhard Rahm. Comparison of schema matching evaluations. In Proc. GI-Workshop "Web and Databases", Erfurt (DE), **2002**.

[11] Peter Brown, Stephen Della Pietra, Vincent Della Pietra, and Robert Mercer. **1993**. The mathematics of machine translation: Parameter estimation. Computational Linguistics, 19(2):263–311, June.

[12] Chris Callison-Burch, David Talbot, and Miles Osborne. **2004**. Statistical machine translation with word- and sentence-aligned parallel corpora. In Proceedings of ACL.

[13] Bies, A. (**2006**). English-Arabic Treebank v 1.0. LDC Cat. No.: LDC2006T10.

[14] Pareto_principle, Princeton University, access online: https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Pareto_principle.html.