Thomas Lund

Brother Rodney Keller

ENG 316

October 27, 2011

Irrefutable Logic: A New Method for Standard Search Protocol

To achieve results more concise, accurate, and relevant than those of current online word reference tools, a degree of programming logic is required. This programming, often called a *logic engine*, will be what distinguishes this new tool from the others. It improves the concept of search query by employing many smaller queries within the larger, main search. The results are gathered, then prioritized or discarded.

The first step in the process is the query. At least two words are used in this method of search. In the **Figure 1** below, two words are submitted for query, one represented in blue and the other in red. These, **Word 1** and **Word** 2 are on the top level. The logic engine generates a list of synonyms for each word. These are illustrated by the boxes on the second level, the dual tone representing the original word with the newly-generated list of **ALL SYNONYMS.**

Both lists of synonyms are compared and cross-referenced. Words that appear in both lists are extracted and compiled. This is essentially a mixing of the two lists. The remaining list is the result of the query and is displayed to the end user.

Simple Data Flow | ForWord

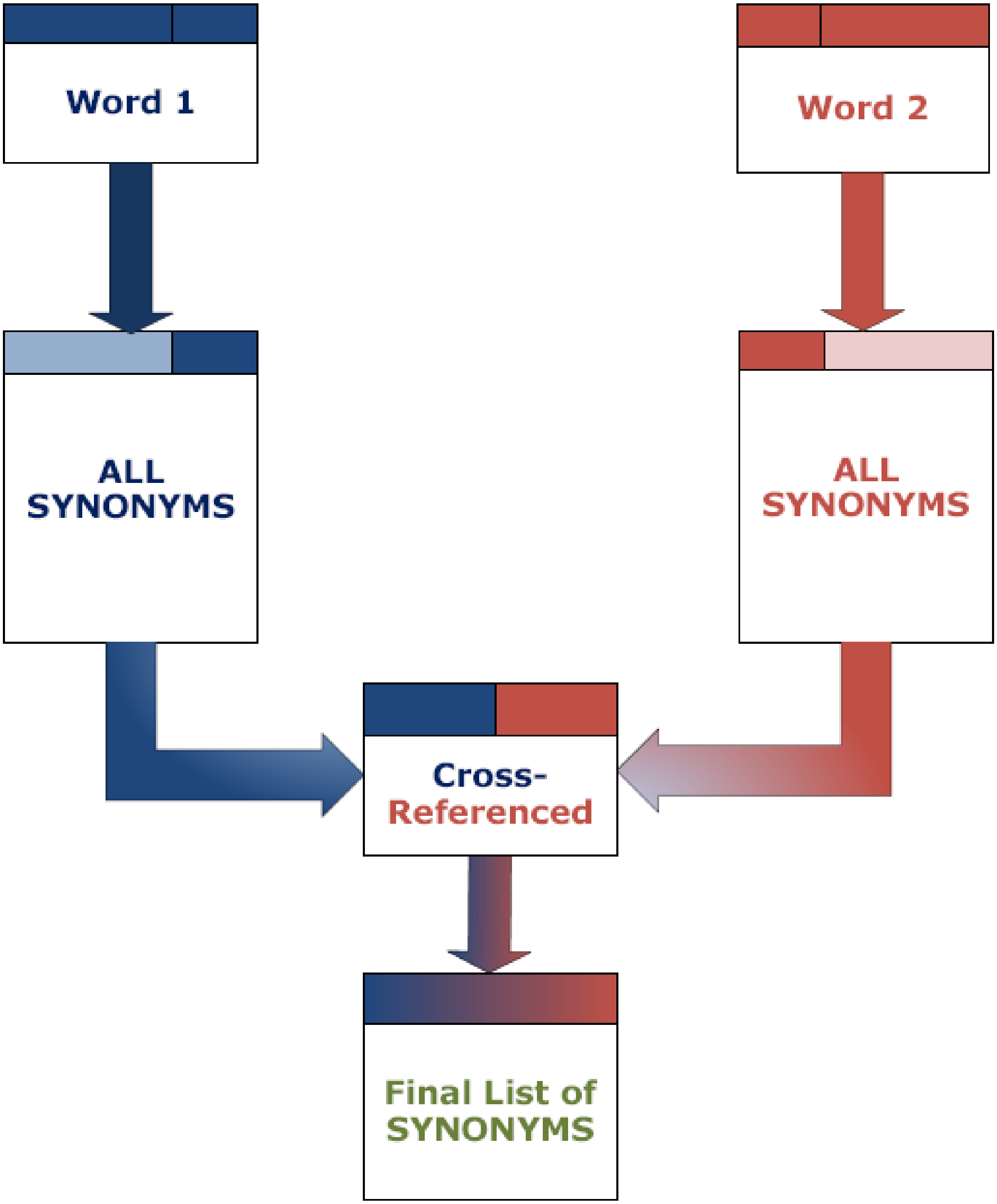


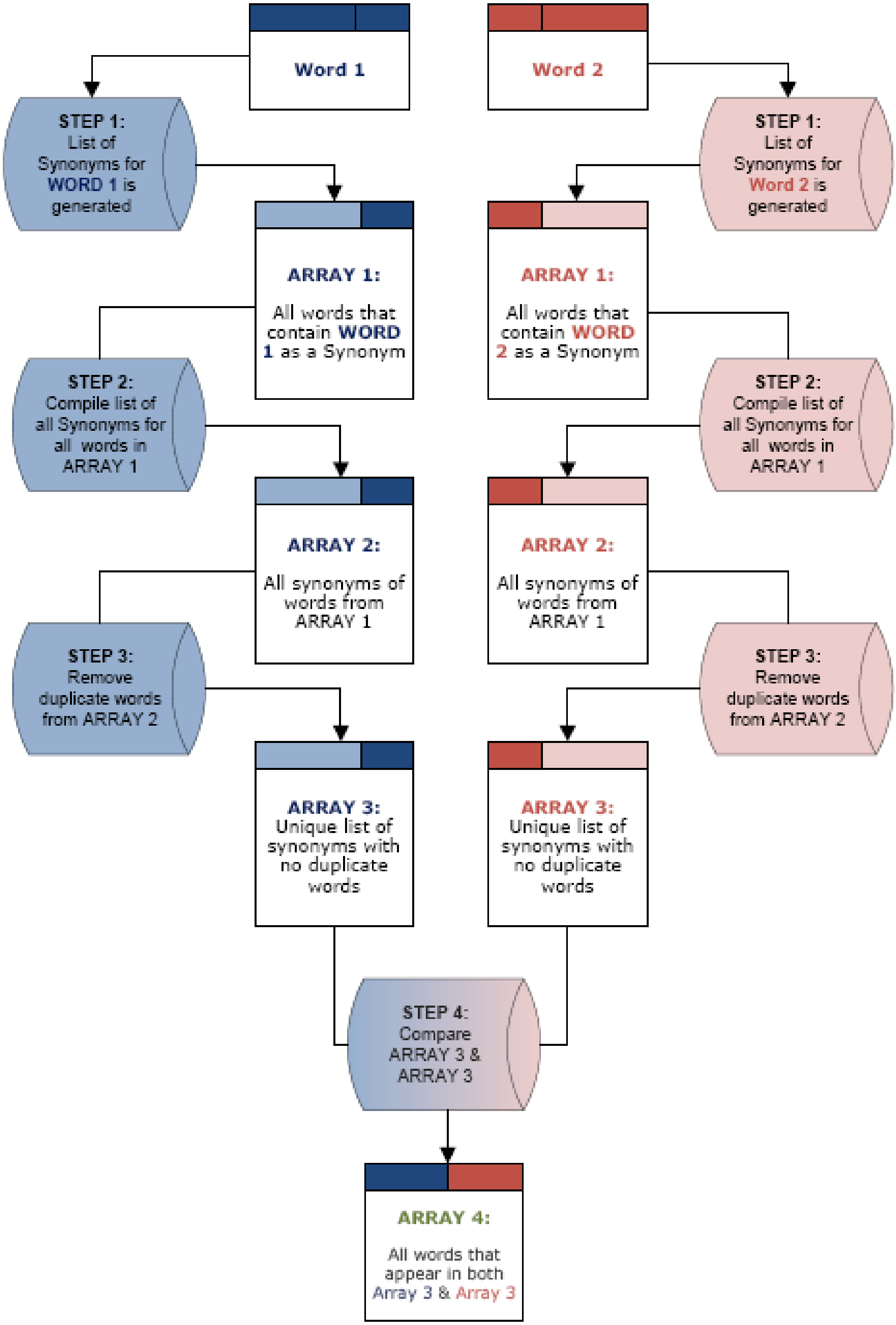
Figure 1

While Figure 1 illustrates the basic query process, a more detailed description of the logic process can be found in Figure 2. Using the same color coding, **Word 1** and **Word 2** and their associated steps and arrays will be separated with the colors blue and red, respectively.

The steps below describe the logic and query process. Please note that two new words are introduced: **database** and **array.** While the database is the resident, central collection of words, the array is the smaller, set of words retrieved from the database. This distinction is important, as the array will be manipulated as the search progresses.

> **STEP 1** compares **Word 1** with the comprehensive database and compiles a list of words that contain the given word as a synonym. This step creates **ARRAY 1**.
>
> **STEP 2** compiles a list of every synonym for ever word is **ARRAY 1**. This step creates **ARRAY 2**.
>
> **STEP 3** removes all duplicate words from **ARRAY 2**. This step creates **ARRAY 3.**
>
> **STEP 4** compares **Word 1, ARRAY 3** and **Word 2, ARRAY 3** and returns all words that appear in both arrays. This step creates **ARRAY 4**, which is used to populate the results page.

Simple Data Flow | ForWord



Word 1
Word 2
STEP 1: List of Synonyms for WORD 1 is generated
STEP 1: List of Synonyms for Word 2 is generated
ARRAY 1: All words that contain WORD 1 as a Synonym
ARRAY 1: All words that contain WORD 2 as a Synonym
STEP 2: Compile list of all Synonyms for all words in ARRAY 1
STEP 2: Compile list of all Synonyms for all words in ARRAY 1
ARRAY 2: All synonyms of words from ARRAY 1
ARRAY 2: All synonyms of words from ARRAY 1
STEP 3: Remove duplicate words from ARRAY 2
STEP 3: Remove duplicate words from ARRAY 2
ARRAY 3: Unique list of synonyms with no duplicate words
ARRAY 3: Unique list of synonyms with no duplicate words
STEP 4: Compare ARRAY 3 & ARRAY 3
ARRAY 4: All words that appear in both Array 3 & Array 3

Figure 2

Described above is effectively an automated cross-referencing process. Creating a system by which an array can be manipulated means a much more thorough search process. The ideal search query would return a list of results considerably more concise than standard online thesauruses, dictionaries, and translators. For example, a cross-referencing search between the words "walk" and "aimless" would generate a shorter list of common synonyms. The list would look something like this:

1. Amble
2. Cruise
3. Drift
4. Float
5. Jaunt
6. Meander
7. Roam
8. Stroll

Though this list of search results is shorter than that of other online thesauruses, it would undoubtedly be more pointed and specific. The benefits of developing such a logic process is essential to the success of the tool.