Capítulo 7

Trabajo a futuro

Aunque el propósito principal de este proyecto era probar la teoría formulaica como un modelo factible para hacer *clustering* sobre los textos, esta tarea depende de otros procesos que están girando alrededor e influyen directamente en los resultados arrojados por el algoritmo. La implementación puede ser mejorada en muchos aspectos y éstos representan en sí mismos opciones de trabajo a futuro.

7.1. Detección de formulaicidad

Uno de los principales problemas para realizar las pruebas es que fue difícil elegir dominios con la certeza de que realmente fuesen formulaicos. No existe una metodología formal para determinar si un dominio es formulaico; creo que esta implementación puede ser útil para hacerlo. Lo anterior se puede lograr definiendo formalmente el concepto de índice de formulaicidad, cuyo valor es numérico y además es directamente proporcional al grado de formulaicidad de un dominio. Calculando el valor de un término como éste, por lo menos se puede tener más certeza y tener bases o una justificación más formal de por qué elegir un dominio sobre otro.

7.2. Tamaño de las fórmulas

Actualmente la implementación obtiene fórmulas de tamaño fijo, de dos palabras; una de las mejoras que se pueden hacer al sistema es implementar la fase de refinamiento de fórmulas definida en la metodología. Esta fase permitirá construir fórmulas de tamaño variable, incluso aplicar el proceso propuesto en [21] para agregar variables a las fórmulas. Este módulo en sí mismo es otro proyecto, no es trivial, aunque el trabajo es mucho menor porque la definición formal ya está incluida en este trabajo. Creo que el problema principal será encontrar una implementación no ingenua para que el tiempo de ejecución no sea tan grande.

También se sugirió un algoritmo en el capítulo de metodología, para la implementación del refinamiento de fórmulas, aunque se podría pensar en algún otro algoritmo.

7.3. Extensión a otros idiomas

La presente implementación sólo es aplicable para textos en español, aunque sólo se requeriría la creación de bases de conocimiento de abreviaturas y palabras vacías e incluir algoritmos de reducción a raíces léxicas para cada uno de los idiomas que se piensen agregar, ya que el diseo y la implementacin del sistema son modulares. El problema con el algoritmo de reducción es que, aunque existen sus versiones en muchos idiomas, en mi experiencia ninguna es una implementación libre de errores, es más, hay algunas que son bastante malas. Lo más recomendable es implementar uno por completo o mejorar alguno de los que se encuentran disponibles. Debido a que la versión del algoritmo de Porter no fue muy efectiva, los resultados de las pruebas realizadas fueron afectados de manera negativa. Seguramente con una mejor implementación de

éste algoritmo se pueden generalizar mucho más las fórmulas. La generalización de la que se habla, es la misma de la que se habló en el capítulo de metodología.

7.4. Definición de términos

No sólo queda trabajo por hacer en el lado de implementación, también en el lado de teoría y formalización. En esta metodología se definieron algunos atributos de las fórmulas: términos como peso, carga semántica y representatividad fueron bastante útiles porque resolvieron algunos problemas aplicando otras teorías lingüísticas. Por ejemplo, la carga semántica pone en práctica la teoría de la relevancia y con esta variable se le da más peso a aquellas fórmulas que contienen menos palabras vacías. También se ideó la fórmula para calcular el índice de similitud.

Se podría pensar en definir otros atributos para las fórmulas; por ejemplo, incluir también a la frecuencia, porque en está implementación sólo se puso en práctica la representatividad; quizás se obtendrían mejores resultados incluyéndola. Además se podría encontrar otra forma para calcular el índice de similitud. Mayor desarrollo de conceptos de formulaicidad en lingstica terica es indispensable para facilitar su aplicacin en lingstica computacional.

7.5. Más intentos con SUBDUE

Aunque en este proyecto no resultó muy bien la experimentación con SUBDUE, creo fuertemente que podría ser útil aunque no de una manera directa: podría explorarse la posibilidad de hacer una extensión. El principal problema con SUBDUE es que no es muy amigable o natural, y como se pudo ver en los experimentos exploratorios, requiere un gran postprocesamiento para finalmente obtener patrones que puedan ser utilizados

para agrupar los textos *clustering*.

Todas las anteriores son opciones reales para continuar este trabajo, y cada una de ellas son interesantes y complejas y por ello requieren bastante trabajo.

7.6. Pruebas con otros córpora

Después de los inconvenientes presentados para la recopilación del algoritmo, sería interesante analizar el comportamiento del algoritmo con otros córpora mejor recopilados. Se ha sugerido que se pruebe el sistema con unas colecciones de prueba del ICT, aún faltaría evaluar estas colecciones para saber si en realidad pueden ser utilizadas.

7.7. Comparaciones con otros trabajos

Comparar los resultados obtenidos con los de los trabajos propuestos a futuro sería interesante y le daría mucho más validez al trabajo presentado en este proyecto. Con las mediciones de tiempo de cada módulo, se puede calcular la complejidad de los algoritmos y con el valor obtenido se tendrá un argumento definitivo que avale las mejoras o avances logrados.

Todas las anteriores son opciones reales para continuar este trabajo, y cada una de ellas es interesante y compleja y por ello requieren bastante trabajo.