https://doi.org/10.33261/jaaru. 2022.29.3.008 *Association of Arab Universities Journal of Engineering Sciences (2022)29(3): 63–73*

Association of Arab Universities Journal of Engineering Sciences

مجلة اتحاد الجامعات العربية للدراسات والبحوث الهندسية

كلية الهندسة - جامعة بغداد

جمعية كليات الهندسة اعضاء اتحاد الجامعات العربية

تحليل معاني جمل اللغة الانكليزية باستخدام خوارزميات التنقيب المحسنة في المعطيات

ميريام شبلي عبيد [1،*]، ***يعرب ديوب*** [2]

[1] *قسم هندسة تكنولوجيا المعلومات، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، سوريا*

[2] *قسم هندسة تكنولوجيا المعلومات، كلية هندسة تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، سوريا*، yaaroub-dayoub@hpu.edu.sy

[*] *الباحث الممثل: ميريام عبيد*، *miriamobied@hotmail.com*

نشر في : 30 ايلول 2022

الخلاصة – يعد تدقيق معاني اللغات الطبيعية من الأهداف الأساسية لعلماء اللغة والمهتمين بعلم اللغات الحاسوبية Computational Linguistics لأنه أصبح من الضروري تدقيق النصوص المكتوبة على الحاسب في مجالات مختلفة. يعرض هذا البحث نموذجاً للتحقق من صحة جمل اللغة الإنكليزية من ناحية المعنى عن طريق توليد قواعد المعنى Semantic Rules من قاعدة بيانات تتضمن الكلمات الأكثر تكراراً في اللغة الإنكليزية، وذلك بالاعتماد على احدى خوارزميات التنقيب في المعطيات وهي خوارزمية FP Growth التي تقوم بتوليد قواعد الترابط Association Rules بين الكلمات ولكن هذه القواعد الناتجة عن الخوارزمية تتجاهل تسلسل الكلمات والذي يعتبر امراً مهماً في تحليل المعنى، لذلك تم تعديل الخوارزمية للحصول على قواعد ترابط تعطي أهمية لتسلسل ورود الكلمات، مع مراعاة الزمن اللازم للحصول على هذه القواعد والذاكرة اللازمة لتخزينها.

الكلمات الرئيسية –" تحليل المعاني، التنقيب في المعطيات، قواعد المعنى، خوارزمية FP Growth، قواعد الترابط".

1. المقدمة

تندرج عملية التحقق من معاني جمل اللغة الإنكليزية ضمن مجال معالجة اللغات الطبيعية NLP Natural Language Processing [1].والذي يعد مجالاً هاماً يربط بين علوم الحاسب وعلم اللغة، ولازالت عملية فهم نص من قبل الحاسب والحكم عليه فيما إذا كان صحيحاً من ناحية المعنى من المشاكل الكبرى التي تواجه التطبيقات المعلوماتية. كما انه لا تتوفر حتى الآن أداة برمجية للتحقق من معاني جمل اللغة الإنكليزية.

ان أحد اهم الأساليب المستخدمة في تحليل المعنى هو التنقيب في المعطيات Data Mining والذي هو عملية اكتشاف المعلومات الموجودة في مجموعة ضخمة من البيانات [5]. وهي آلية تهدف الى تحليل كميات كبيرة من البيانات لاستخراج نماذج وقواعد مهمة منها لم تكن معروفة مسبقاً والتي لا يمكن اكتشافها بالطرق التقليدية بسبب ضخامة حجم البيانات أو العلاقات المعقدة جداً بين البيانات [17]. ولتطبيق تقنيات التنقيب في المعطيات نحتاج الى:

- توفر قاعدة بيانات ضخمة تتضمن بيانات عن المسألة المراد حلها.
- اختيار وتطبيق خوارزمية تناسب المسألة المطروحة.

2. الدراسات السابقة

العديد من الأبحاث السابقة خاضت تجربة تحليل المعنى في مجالات مختلفة، مثلاً في مجال قواعد البيانات تم استخدام تحليل المعنى لتوليد مخطط Entity Relationship Model (ER) من وثيقة توصيف البيانات المخزنة في قاعدة البيانات [11]. كما أُستخدم تحليل المعنى في مجال المنشورات العلمية لاكتشاف اتجاهات البحث التي تتطور بسرعة [12]، وفي مجال خدمات الويب [8] وغيرها الكثير من المجالات.

وقد تم استخدام تقنيات مختلفة لتحليل المعنى، فبعض الباحثين قاموا ببناء أنظمة لتحليل المعنى باستخدام تقنية التنقيب في النصوص Text Mining لاستخلاص قواعد الترابط بين الكلمات، ولكن هذه الأنظمة تتجاهل تسلسل ورود الكلمات في النص، وتركز على الكلمات المهمة فقط وعلى توزيعها الاحصائي في النص [2].

واستخدم باحثون آخرون تقنيات التنقيب في المعطيات لبناء نموذج لتحليل المعنى يتحقق من معاني جمل اللغة الإنكليزية الصحيحة قواعدياً، باستخدام خوارزميات التنقيب في المعطيات التي تهتم بإيجاد قواعد الترابط بين البيانات مثل خوارزميتي Apriori و FP Growth، وأبرزوا محاسن وعيوب هاتين الخوارزميتين دون انجاز أي تحسين عليهما، حيث وجدوا ان خوارزمية FP Growth اسرع من Apriori في توليد المجموعات المتكررة، بالمقابل فإن Apriori لا تتطلب مساحة ذاكرة إضافية للتخزين كما في FP Growth [18].

في حين قام باحثون آخرون باستخدام المنطق الطبابي fuzzy logic في إيجاد المعنى المقصود للكلمة في جملة معينة بالاعتماد على قاعدة بيانات معجمية تربط الكلمات من خلال علاقات مختلفة وإعطاء وزن للعلاقات بين الكلمات حسب أهميتها، وتمثيل ذلك في مخطط ضبابي Fuzzy graph [15]. ولجأ آخرون الى الشبكات العصبونية لبناء نموذج لتحليل جودة النص المكتوب باللغة الإنكليزية بالاعتماد على شبكة عصبونية متكررة Recurrent Neural Network [8]، ولكن ابحاثهم كانت تستند لنتائج تجريبية فقط.

في هذا البحث تم الاعتماد على تقنيات التنقيب في المعطيات، وهي خوارزمية FP Growth التي تم تطبيقها على قاعدة بيانات تتضمن الكلمات المترافقة باللغة الإنكليزية، للحصول على قواعد المعنى المناسبة لاستخدامها في التحقق من صحة معاني جمل اللغة الإنكليزية، لكن قواعد المعنى الناتجة عن هذه الخوارزمية لا تراعي تسلسل ورود الكلمات، الذي يعتبر امراً مهما في تحليل المعنى، لذا تم اقتراح تعديل على خوارزمية FP Growth للحصول على قواعد معنى تعطي أهمية لتسلسل ورود الكلمات، مع مراعاة الزمن اللازم للحصول على هذه القواعد والذاكرة اللازمة لتخزينها.

3. اهداف البحث واهميته

يهدف هذا البحث إلى تحليل معاني جمل اللغة الإنكليزية والتحقق من صحتها، وذلك بالاعتماد على خوارزمية FP Growth وهي احدى خوارزميات التنقيب في المعطيات التي تهتم بإيجاد الترابط بين البيانات.

تأتي أهمية هذا البحث في انه يقترح آلية لمعالجة اللغات الطبيعية وبناء أنظمة تحاكي الانسان بمجال اللغة، كما انه يعد خطوة مهمة لتطبيقه في مجالات مختلفة مثل تدقيق رسائل البريد الالكتروني والمقالات والأبحاث العلمية وتطوير عملية التصحيح في الامتحانات الإلكترونية وكما انه يوفر أداة تعليمية مفيدة للغات الطبيعة.

4. منهجية البحث

تم اولاً تحديد المسألة المراد حلها وهي إيجاد قواعد المعنى لاستخدامها في التأكد من صحة جمل اللغة الانكليزية من ناحية المعنى، وتحديد المتطلبات اللازمة لحل هذه المسألة كما يلي:

4.1 قاعدة بيانات تتضمن كلمات اللغة الإنكليزية:

تم الحصول على قاعدة بيانات ضخمة تتضمن بيانات عن كلمات اللغة الانكليزية من الموقع The Corpus of Contemporary American English لجامعة Brigham Young [3]، والذي يتضمن ملفات نصية تحتوي كلمات متسلسلة باللغة الإنكليزية مع الصنف الاعرابي لكل كلمة مثل (فعل، اسم، صفة، حرف جر....)، هذه الملفات تتضمن حوالي مليون سجل للكلمات المترافقة الأكثر تكراراً في اللغة الإنكليزية لكل من الحالات (كلمتين متسلسلتين، ثلاث كلمات متسلسلة، أربع كلمات متسلسلة، خمس كلمات متسلسلة) . يوضح الجدول (1) عينة عشوائية من الملف الذي يحتوي ثلاث كلمات متسلسلة.

الجدول 1: عينة عشوائية من الملف الذي يحتوي ثلاث كلمات متسلسلة

word1	word2	word3	pos1	pos2	Pos3
deep	blue	sea	jj	jj	nn1
eat	and	drink	vv0	cc	vv0
went	on	sale	vvd	ii	nn1

يحتوي الجدول السابق على الاعمدة التالية:

- الاعمدة **word1** و**word2** و**word3** تتضمن الكلمات الثلاثة حسب تسلسل ورودها.
- الاعمدة **pos1** و**pos2** و**Pos3** تتضمن الصنف الاعرابي للكلمات الأولى والثانية والثالثة على التوالي.

يمكن توضيح الاختصارات الممثلة للصنف الاعرابي لبعض الكلمات، بالجدول (2) مرتبة حسب الترتيب الابجدي:

الجدول 2: الاختصارات الممثلة للصنف الاعرابي

الرقم	الرمز	التوصيف
1	cc	اداة ربط coordinating conjunction
2	ii	حرف جر عام general preposition
3	jj	صفة عامة general adjective
4	nn1	اسم مفرد singular noun
5	vv0	الشكل الأساسي للفعل base form of lexical verb
7	vvd	الزمن الماضي للفعل past tense of lexical verb

تم اضافة الملفات التي تحتوي على الكلمات المترافقة باللغة الانكليزية الى قاعدة بيانات MySQL، وبعد ذلك تم إنشاء قاعدة بيانات تتضمن جداول توافق خمس مجموعات مهمة لقواعد المعنى بالاعتماد على الملفات السابقة، حيث تم تنظيم الجداول بالتسلسل التالي:

- الجدول Adjective_Noun يخزن الصفات وما يليها من أسماء.
- الجدول Noun_Verb يخزن الأسماء وما يليها من أفعال.
- الجدول Verb_Noun يخزن الأفعال وما يليها من أسماء.
- الجدول Verb_Preposition يخزن الأفعال وما يليها من أحرف جر.
- الجدول Preposition_Noun يخزن أحرف الجر وما يليها من أسماء.

علماً أنه يمكن إضافة او حذف جداول أخرى وذلك حسب الحاجة، حيث سيتم لاحقاً تطبيق احدى خوارزميات التنقيب في المعطيات على الجداول السابقة للحصول على قواعد المعنى الملائمة، وبما أن الجداول التي تم الحصول عليها من الموقع تتضمن حوالي مليون سجل للكلمات المترافقة والذي يعتبر عدداً كبيراً من السجلات، بالتالي زمن البحث فيها سيكون كبيراً جداً لذلك تم تقسيمها الى خمس جداول تعتبر مهمة من ناحية المعنى.

4.2 الآلية المقترحة للتحقق من صحة الجملة:

قبل التحقق من صحة الجملة من ناحية المعنى يجب التحقق اولاً من صحتها مفرداتياً ثم قواعدياً، وليس بالضرورة أن تكون كل جملة صحيحة قواعدياً هي جملة صحيحة من ناحية المعنى، فمثلاً الجملة The car eats the apple هي جملة صحيحة قواعدياً ولكنها غير صحيحة من ناحية المعنى.

حيث يتم التحقق من صحة الجملة مفرداتياً باستخدام قاموس (جدول) يتضمن كلمات اللغة الإنكليزية مع نوع كل كلمة (فعل، اسم، صفة، حرف جر....) وذلك للتأكد من ان مفردات(كلمات) الجملة هي ضمن كلمات اللغة الإنكليزية، والجدول التالي يمثل عينة عشوائية من القاموس المستخدم:

الجدول 3: عينة من القاموس المستخدم

number	word	type
1	a	Determiner
2	Abandon	Verb
3	Abandoned	Adjective
4	Abandonment	Noun

ثم يتم التحقق من صحة الجملة نحوياً باستخدام النحو الشكلي Formal Grammar، والذي هو مجموعة من الأسس والقواعد التي تبين طريقة تكوين عبارات اللغة البسيطة والمركبة [9]، ويرمز له بالشكل $G=(V_N,V_T,P,S)$ حيث:

V : مجموعة مفردات اللغة والتي يمكن توليدها باستخدام النحو الشكلي G وهي نوعان:

$\mathbf{V_N}$: non_terminal symbols مجموعة العناصر غير النهائية وتمثل العقد الداخلية في شجرة النحو.

V_T: terminal symbols مجموعة العناصر النهائية وتمثل الأوراق في شجرة النحو.
حيث $V_T \cap V_N = \Phi$ و $V_T \cup V_N = V$

P: productions rules مجموعة قواعد الاشتقاق، ولها الصيغة $A \rightarrow \beta$.
S: start symbol محرف (عنصر) البداية ويمثل جذر الشجرة، حيث $S \epsilon V_N$.

وهناك عدة أنواع للنحو الشكلي، سيتم استخدام النحو الحر context –free formal grammar، حيث ان اللغات المولدة عن هذا النحو تدعى باللغات type2 او context-free langue لأنها لا تتعلق بحيز ما، فلا توجد قيود على نوعية المحارف الواقعة على يمين السهم في القواعد المولدة فانه يمكن بالتالي توليد لغات عملاقة ذات مزايا واسعة جدا كافية لتوصيف وتحليل أعقد المسائل العلمية واللغات الخاصة بالتعرف على الصور والصوت والاشكال والخ [9].

وقد تم بناء نموذج نحوي للتحقق من صحة الجملة نحوياً، حيث تمثل القواعد التالية في الجدول (4) النموذج النحوي المستخدم، فإذا كانت سلسلة مفردات الجملة متطابقة مع احدى قواعد النموذج النحوي تكون الجملة صحيحة نحوياً والا تعتبر غير صحيحة نحوياً.

الجدول 4: النموذج النحوي المستخدم

S → <NP> <VP> |<NPP> <VP> |<VP> |<NP> <NPP> <VP> |<NPP> <NPP> <NP> <VP>
NP → <N> |<Det> <Adj> <N> |<Det> <N> |<Pron> |<Pron> <N> |<Num> <N> |<Num> <N> <N> |<N> <Conj> <N> |<Num> <N> <N> <Conj> <N> |<Det> <N> <N> |<Det> <Adj> <Adj> <N> |<Pron> <N> <N> |<Adj> <Pron> <N> |<Det> <Adj> <N> <N> |<Det> <Adj> <N> <Pron> |<Neg> <N> |<Pron> <Adj> <N>
NPP → <Prep> <NP>
AP → <Adj> |<Adj> <Adj> |<Adj> <Conj> <Adj>
APP → <Prep> <AP>
V → <V> |<V> <V> |<V> <Adv> <V> |<V> <Neg> <V> |<V> <V> <V> <V> |<V> <Conj> <V> |<V> <Adv> |<V> <Neg> <V> <Adv> |<Adv> <Conj> <Adv> |<Adv> <V> <Neg> <V> |<V> <Adv> <Conj> <Adv> |<Adv> <V> |<V> <V> <Adv>
VPP → <Prep> <V>
VP → <V> <NP> |<V> <VPP> <NP> |<V> <NPP> <NP> |<V> <NP> <NPP> |<V> <AP> |<V> <NP> <NP> <VPP> |<V> |<V> <NPP> |<V> <VPP> |<V> <NP> <V> |<V> <NP> <VPP> <NP> |<V> <VPP> <NPP> |<V> <NP> <NPP> <V> <NP> |<V> <NP> <AP> |<V> <NP> <AP> <VPP> |<V> <NPP> <NPP> |<V> <NP> <V> <NPP> |<V> <VPP> <NP> <NP> |<V> <NP> <NPP> <NPP> |<V> <NPP> <NPP> <NPP> |<V> <VPP> <AP> <NPP> <NPP> |<V> <VPP> <NP> <NPP> |<V> <AP> <NPP> <NPP> |<V> <NP> <AP> <NPP> |<V> <NPP> <AP> |<V> <VPP> <NP> <AP> |<V> <AP> <NPP> |<V> <NP> <VPP> <NP> <NPP> |<V> <NP> <NPP> |<V> <NPP> <VPP> <NP> |<V> <NPP> <AP> <NPP>

تم توضيح الرموز المستخدمة في النموذج النحوي السابق، في الجدول (5) التالي:

الجدول 5: الاختصارات المستخدمة في النموذج النحوي

المعنى	الاختصارات
Sentence	S
Determiner	Det
Adjective	Adj
Pronoun	Pron
Numerals	Num
Conjunction	Conj
Negation	Neg
Preposition	Prep
Adverb	Adv
Verb	V
Verb Command	VC
Noun	N
Noun Phrase	NP
Verb Phrase	VP
Adjective Phrase	AP
Noun Preposition Phrase	NPP
Verb Preposition Phrase	VPP
Adjective Preposition Phrase	APP

بعد التحقق من صحة الجملة مفرداتياً ونحوياً، يتم التحقق من صحة معاني الجملة باستخدام قواعد المعنى الناتجة عن خوارزمية التنقيب في المعطيات والتي سنقوم لاحقاً بتوضيحها، والآن سيتم توضيح الآلية المقترحة للتحقق من صحة الجملة بواسطة مثال، لتكن لدينا الجملة التالية:

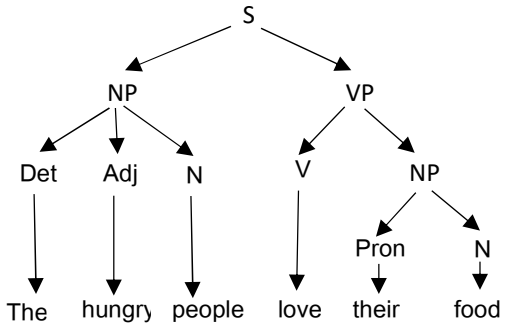
The hungry people love their food

يتم التحقق من صحة الجملة وفق الخطوات التالية:

1- التحقق اولاً من صحة الجملة مفرداتياً وتحديد الصنف الاعرابي لكل كلمة، عن طريق قاموس كلمات اللغة الإنكليزية الموضح في الجدول(3) :

The**(Determiner)** hungry**(adjective)** people**(noun)** love**(verb)** their**(Pronoun)** food**(noun)**

2- التحقق من صحة الجملة نحوياً عن طريق النموذج النحوي السابق الموضح في الجدول(4) وبناء شجرة النحو للجملة كما يلي:



الشكل1: شجرة النحو للمثال

3- حذف الكلمات غير المهمة من ناحية المعنى:
مثل أدوات الربط بين أجزاء العبارة وأدوات التعريف والضمائر الشخصية، في مثالنا هذا تم حذف أداة التعريف the والضمير their.

hungry people love food

4- مناقشة جميع حالات الكلمات المترابطة:

- التحقق من وجود الكلمتين hungry people ضمن مجموعة قواعد المعنى Adjective → Noun
- التحقق من وجود الكلمتين people love ضمن مجموعة قواعد المعنى Noun → Verb
- التحقق من وجود الكلمتين love food ضمن مجموعة قواعد المعنى Verb → Noun

فعند إيجاد الكلمات السابقة جميعها ضمن قواعد المعنى، عند ذلك يتم التأكد من صحة الجملة من ناحية المعنى.

4.3 دراسة خوارزميات التنقيب لتنفيذ التحليل المعنوي:

لحل المسألة تم دراسة عدة خوارزميات تنقيب في المعطيات تختص بإيجاد الترابط بين البيانات مثل Apriori وFP Growth و Vertical data format:

تستخدم خوارزمية Apriori طريقة التوليد والاختبار generate and test، أي انها تولد مجموعة من العناصر(الكلمات) المرشحة ثم تختبر فيما إذا كانت تمثل عناصر متكررة، بالتالي فإن هذه الخوارزمية تستغرق زمناً كبيراً بالإضافة الى مساحة تخزينية كبيرة، خاصة إذا كانت قاعدة البيانات ضخمة وكان عدد العناصر المرشحة المختبرة كبيراً [10].

تعتمد خوارزمية Vertical data format [5] على مبدأ تحويل الاسطر في الجدول الى أعمدة، أي أنه من أجل كل عنصر في قاعدة البيانات يتم تخزين قائمة بأرقام الاسطر التي يتواجد فيها العنصر، وبذلك يتم تمثيل البيانات بشكل عمودي، وبعد ذلك يتم حساب مجموعات العناصر المتكررة. تعتبر هذه الخوارزمية سريعة لكن القوائم الوسيطية التي تضم أرقام أسطر العناصر قد تكون ضخمة جداً خاصة إذا كانت قاعدة البيانات كبيرة، وبالتالي تتطلب زمن تنفيذ ومساحة تخزينية أكبر، مما يحد من استخدامها [4].

يقوم المبدأ الأساسي لخوارزمية FP) Frequent Pattern Growth) [5] على استراتيجية فرق تسد divide and conquer، فتحول قاعدة البيانات الضخمة الى بنية شجرية حجمها أصغر من حجم قاعدة البيانات الاصلية، تسمى شجرة النموذج المتكرر frequent pattern tree (اختصاراً FP tree)، وتعتبر تطويراً هاماً لخوارزمية Apriori لأنها تعتمد على التنقيب في الشجرة للحصول على العناصر المتكررة دون توليد عناصر مرشحة، واذا كانت قاعدة البيانات كبيرة فإن بناء شجرة FP tree قد يستغرق وقتاً، لكن بمجرد بنائها تتم قراءة العناصر المتكررة بسهولة [14, 18].بذلك سيتم الاعتماد على خوارزمية FP Growth في البحث.

4.4 مبدأ عمل خوارزمية *FP Growth*:

قبل شرح مبدأ عمل الخوارزمية سنوضح بعض المفاهيم الأساسية [6]:

- **الدعم الأصغري minimum_ support** (min_sup): هو ثابت عددي موجب، أكبر او يساوي الواحد، يتم تحديده من قبل المستخدم، حيث يعرّف الدعم لعنصر ما بأنه عدد الاسطر في قاعدة البيانات التي تتضمن هذا العنصر.
- **العنصر المتكرر frequent item**: نقول عن عنصر أنه عنصر متكرر إذا وفقط إذا كان تكرار هذا العنصر أكبر أو يساوي الدعم الأصغري min_sup.

لتكن $I=\{a_1, a_2, \dots a_m\}$ مجموعة من العناصر، و $DB=\{T_1,T_2,\dots, T_n\}$ مجموعة تمثل اسطر قاعدة البيانات، حيث $(i \in [1\text{-}n])$ T_i هو سطر في قاعدة البيانات يحتوي على مجموعة من العناصر من I، فعلى سبيل المثال ان الدعم للعنصر a_1 هو عدد الاسطر في قاعدة البيانات التي تحتوى على العنصر a_1، ويكون العنصر a_1 متكرراً اذا كان دعم العنصر a_1 أكبر او يساوي الدعم الاصغري min_sup المحدد مسبقاً.

- **شجرة النموذج المتكرر frequent pattern tree (اختصاراً FP tree):** هي بنية شجرية مكونة من جذر واحد يسمى root، ومجموعة من الأشجار الفرعية الأبناء للجذر التي تتكون من عقد والروابط بين العقد، حيث كل عقدة تتكون من اسم العنصر الذي تمثله العقدة، وعداد يمثل عدد مرات تكرار القعدة.

خطوات عمل الخوارزمية [5]:

الدخل: DB قاعدة البيانات، min_sup الدعم الاصغري.

الخرج: قواعد الترابط (قواعد المعنى).

الطريقة:

1. المرور الأول على قاعدة البيانات لإيجاد مجموعة تتضمن كل عنصر مع تكراره وتخزينها في اللائحة L.
2. ايجاد العناصر المتكررة frequent items التي تكرارها أكبر او يساوي الدعم الاصغري.
3. ترتيب العناصر المتكررة في اللائحة L ترتيباً تنازلياً بحسب قيمة تكرارها.
4. المرور الثاني على قاعدة البيانات لبناء شجرة النموذج المتكرر FP tree كما يلي:
   a. انشاء جذر الشجرة وليكن root.
   b. ترتيب العناصر المتكررة في كل سطر من أسطر قاعدة البيانات بحسب الترتيب في اللائحة L، أي بحسب قيمة تكرارها تنازلياً، ولتكن قائمة العناصر المتكررة في السطر T هي [P|S] حيث P هو العنصر الأول وS هو القائمة المتبقية.
   c. انشاء فرع في الشجرة لكل سطر من أسطر قاعدة البيانات، كما يلي: إذا كانت الشجرة تتضمن عقدة N لها الاسم نفسه للعنصر الحالي في قاعدة البيانات اي N.item-name = P.item-name يتم زيادة عداد العقدة N بمقدار واحد، والا يتم انشاء عقدة جديدة N يكون العداد فيها مساوياً الواحد مع ربطها مع العقدة الاب لها، ويتم تكرار الآلية السابقة حتى تصبح القائمة S فارغة.
5. التنقيب في الشجرة للحصول على قواعد الترابط: يتم التنقيب في الشجرة عن طريق تحديد المسارات المرتبطة في شجرة FP tree لكل عقدة وهو ما يسمى قاعدة النموذج الشرطي Conditional Pattern Base، وبناءً على المسارات التي تم ايجادها في قاعدة النموذج الشرطي يتم بناء شجرة FP الشرطية (Conditional FP tree) لكل عقدة، حيث يتم اخذ العناصر المتكررة فقط (التي تكرارها أكبر او يساوي الدعم الاصغري)، ومن ثم يتم الحصول على قواعد الترابط من خلال النماذج المتكررة المولدة من شجرة FP الشرطية.

اما التعقيد الزمني للخوارزمية فهو يعتمد على البحث عن المسارات في شجرة FP tree، بالتالي التعقيد الزمني للبحث عن جميع المسارات [13]، موضح بالعلاقة:

O(frequent item count ^{2} * maximum depth of tree).

حيث:frequent item count : عدد العناصر المتكررة في قاعدة البيانات.
maximum depth of tree: العمق الاعظمي لشجرة FP tree.

4.4.1 مثال عملي على خوارزمية FP Growth:

سيتم توضيح مبدأ عمل خوارزمية FP Growth من خلال المثال البسيط التالي، سيتم تطبيق الخوارزمية على الجدول (6) الذي يتضمن أربع أسطر لتتالي فعل واسم، وذلك من أجل دعم اصغري min_sup =1.

الجدول 6: البيانات المخزنة

ID	Verb	Noun
1	eat	food
2	find	food
3	need	food
4	need	work

عند المرور الأول على الجدول (6) يتم تشكيل اللائحة L (الجدول (7))، وهي مجموعة من العناصر المتكررة مع تكرارها بعد ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها.

الجدول 7: لائحة العناصر المتكررة عند تطبيق خوارزمية FP Growth

Item_id	support
food	3
need	2
eat	1
find	1
work	1

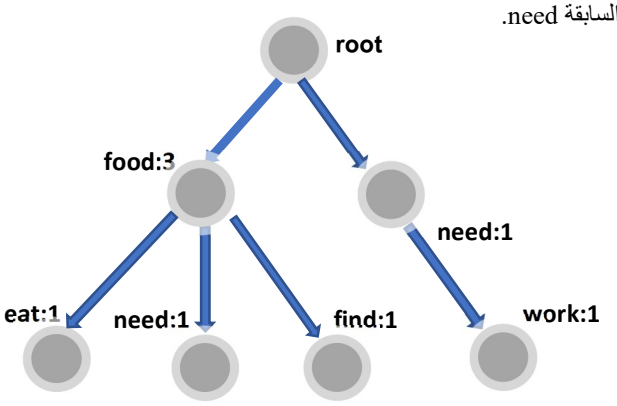
عند المرور الثاني على الجدول (6) يتم بناء شجرة العناصر المتكررة FP tree الموضحة بالشكل(2)، بالآلية التالية:

من اجل السطر الأول في قاعدة البيانات (الجدول(6)) يتم ترتيبه بحسب لائحة العناصر المتكررة L فيصبح من اليسار الى اليمين food,eat لان العقدة food تكرارها 3 اكبر من تكرار العقدة eat ، ثم يتم بناء الفرع الأول في الشجرة الذي يتألف من عقدتين، العقدة الأولى food:1 تكرارها 1 وتكون عقدة ابن لجذر الشجرة، العقدة الثانية eat:1 ترتبط بالعقدة food.

من اجل السطر الثاني يتم ترتيبه فيصبح food,find نلاحظ انه يشترك بالبادئة food مع المسار السابق، لذلك لا نقوم بإضافة عقدة جديدة food بل نقوم بزيادة عداد العقدة الموجودة food بمقدار 1 لتصبح food:2 ثم يتم انشاء عقدة جديدة find:1 ترتبط بالعقدة food.

من اجل السطر الثالث يتم ترتيبه فيصبح food,need كذلك يشترك بالبادئة food مع المسار السابق، فيتم زيادة عداد العقدة food بمقدار واحد لتصبح food:3 ثم يتم انشاء عقدة جديدة need:1 ترتبط بالعقدة food.

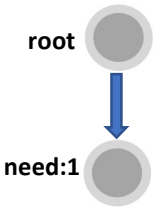
من اجل السطر الرابع والأخير يبقي ترتيبه need,work وبما انه لا يوجد بادئة مشتركة في الشجرة يتم انشاء فرع جديد يتألف من عقدتين، العقدة الأولى need:1 تكون عقدة ابن لجذر الشجرة، والعقدة الثانية work:1 ترتبط بالعقدة السابقة need.



الشكل2: شجرة FP tree بعد تطبيق خوارزمية FP Growth

أخيراً يتم التنقيب في شجرة FP tree كما يلي:

من أجل العقدة work، تم إيجاد مسار واحد لها ضمن الشجرة هو {need:1} وهذا المسار يشكل قاعدة النموذج الشرطي للعقدة work، ثم يتم بناء شجرة FP tree الشرطية للعقدة work بناءً على المسار السابق بالتالي فهي تتضمن مساراً واحداً هو need:1 (لم يتم حذف أي مسار عند بناء شجرة FP tree الشرطية لان الدعم الأصغري يساوي الواحد)، والشكل التالي يمثل شجرة FP tree الشرطية للعقدة work:



الشكل 3: شجرة FP tree الشرطية للعقدة work

من خلال هذا المسار الوحيد في شجرة FP tree الشرطية يتم توليد مجموعات النماذج المتكررة للعقدة work وهي {need, work} وهكذا يتم التنقيب في الشجرة بالنسبة لبقية العقد كما هو موضح في الجدول(8):

الجدول 8: التنقيب في شجرة FP tree

item	Conditional pattern base	Conditional FP tree	Frequent patterns generated
work	{need:1}	need:1	{need, work}
need	{food:1}	food:1	{ food, need }
find	{food:1}	food:1	{ food, find }
eat	{food:1}	food:1	{ food, eat }

من خلال التنقيب في الشجرة نحصل على قواعد الترابط التالية:

if word1='food' **then** word2='eat'
if word1='food' **then** word2='find'
if word1='food' **then** word2='need'
if word1='need' **then** word2='work'

نلاحظ أن بعض هذه القواعد تعتبر قواعد معنوية خاطئة ومعكوسة، مما يؤدي لزيادة زمن التحليل المعنوي وهدر بحجم الذاكرة، فقواعد الترابط السابقة أظهرت الترابط بين الكلمات، لكنها تجاهلت تسلسل ورود الكلمات، لان هذه الخوارزمية تركز على ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها، هذا يعني أنه كلما كانت الكلمة أكثر تكراراً كلما كانت أقرب إلى جذر الشجرة، لكن موضوع ترتيب الكلمات يعتبر أساسيا في هذا البحث لأنه يؤثر على إعطاء المعنى الصحيح للجملة، لذا اصبح من الضروري تعديل الخوارزمية للحصول على قواعد ترابط تعطي أهمية لتسلسل ورود الكلمات.

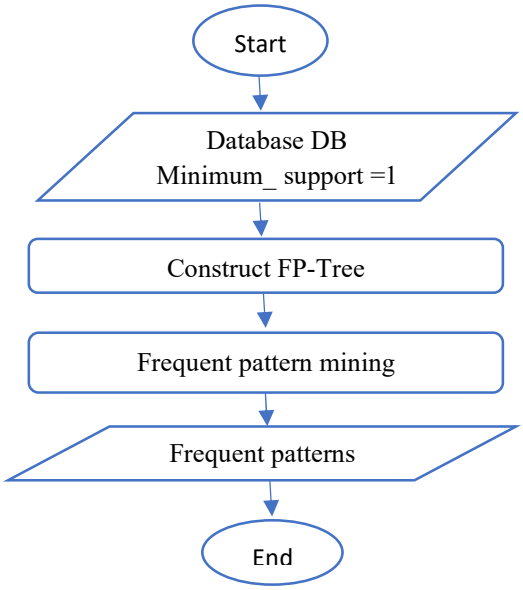
4.5 الخوارزمية المقترحة:

اقترحنا تعديل خوارزمية FP Growth من اجل الحصول على قواعد ترابط تراعي تسلسل ورود الكلمات، حيث كان التعديل على الخوارزمية من خلال حذف الخطوة (ترتيب العناصر المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها)، وفرض قيد على قيمة الدعم الأصغري بحيث تكون قيمته تساوي الواحد، والهدف من هذا القيد هو عدم حذف قواعد الترابط التي تتضمن كلمات يكون تكرارها أقل من قيمة الدعم الأصغري، لأن حذف بعض قواعد الترابط يؤدي الى اعتبار الجمل التي تتضمن واحداً او أكثر من القواعد المحذوفة غير صحيحة من ناحية المعنى على الرغم من وجودها ضمن قاعدة البيانات، كما ان الدعم الاصغري هو قيمة تحدد من قبل المستخدم تجريبياً، مما يجعل النتائج محفوفة بالأخطاء، وبالتالي لا جدوى من تنفيذ الخطوات (1 و 2 و3 و 4-b) من خوارزمية FP Growth وتم اقتراح حذفها، كذلك في الخطوة 5 (التنقيب في شجرة FP tree) سيتم الحصول على قواعد الترابط ممثلة بفروع شجرة FP tree (قواعد النموذج الشرطيConditional Pattern Base للعقد الأوراق فقط) دون بناء أشجارFP الشرطية، لأنه طالما تم فرض قيد على قيمة الدعم الاصغري انه يساوي الواحد بالتالي لا جدوى من بناء أشجارFP الشرطية، بل انه يسبب هدر في الزمن والذاكرة، فتصبح خطوات تنفيذ الخوارزمية المقترحة كالآتي:

1.المرور على قاعدة البيانات لبناء شجرة FP tree كما يلي:

a. انشاء جذر الشجرة وليكن root.
b. انشاء فرع في الشجرة لكل سطر من أسطر قاعدة البيانات، حيث تم الحفاظ على العناصر في كل سطر دون تغيير ترتيبها، فيتم انشاء الفرع كما يلي: إذا كانت الشجرة تتضمن عقدة لها الاسم نفسه للعنصر الحالي في قاعدة البيانات لا يتم انشاء عقدة جديدة، والا يتم انشاء عقدة جديدة مع ربطها مع العقدة الاب لها.

2.التنقيب في شجرة FP tree للحصول على قواعد الترابط ممثلة بفرع شجرة FP tree. والشكل التالي يمثل المخطط التدفقي للخوارزمية المقترحة:

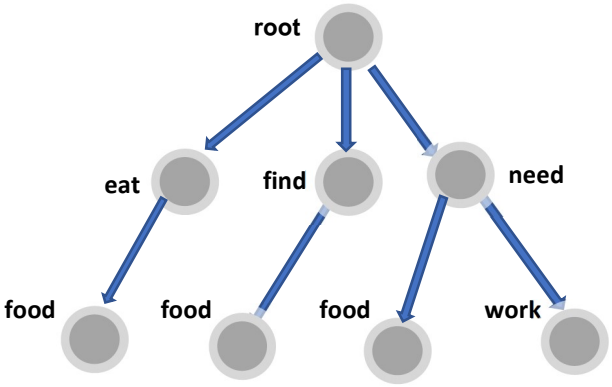


الشكل4: المخطط التدفقي للخوارزمية المقترحة

كما هو ملاحظ لم يتم ذكر عدد مرات تكرار كل عقدة في الشجرة، لأنه في موضوع تحليل المعاني لا يهم عدد مرات تكرار كل كلمة، وهذا يقلل من المساحة التخزينية اللازمة للتخزين.

4.5.1 مثال عملي على الخوارزمية المقترحة:

سيتم تطبيق الخوارزمية المقترحة من أجل المثال السابق نفسه في الجدول (6)، بالتالي سيتم المرور على الجدول مباشرة لبناء شجرة FP tree الموضحة بالشكل(5) كما يلي:



الشكل5: شجرة FP tree بعد تطبيق الخوارزمية المقترحة

ثم يتم التنقيب في شجرة FP tree للحصول على قواعد الترابط التالية الممثلة بفروع شجرة FP tree:

if word1='eat' **then** word2='food'
if word1='find' **then** word2='food'
if word1='need' **then** word2='food'
if word1='need' **then** word2='work'

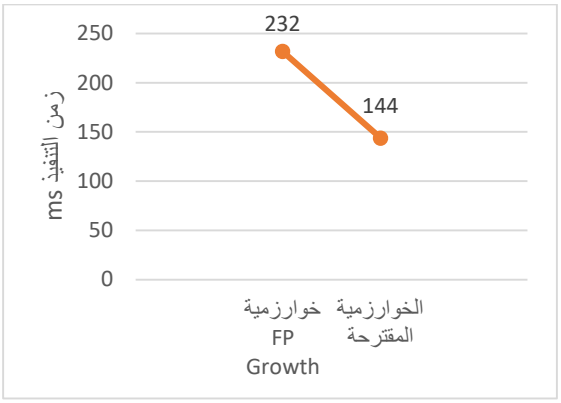
كما هو ملاحظ أن قواعد الترابط السابقة تعتبر قواعد معنوية صحيحة وقد أعطت أهمية لتسلسل ورود الكلمات، ولكن في الشجرة الناتجة عن الخوارزمية المقترحة في الشكل(5) عدد العقد 8 وعدد خطوط الارتباط 7 وكلاهما أكبر من عدد العقد وعدد خطوط الارتباط في الشجرة الناتجة عن خوارزمية FP Growth في الشكل(2)، بالتالي فإن بناء الشجرة يتطلب زمن تنفيذ أكبر وذاكرة تخزينية أكبر.

4.5.2 مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة:

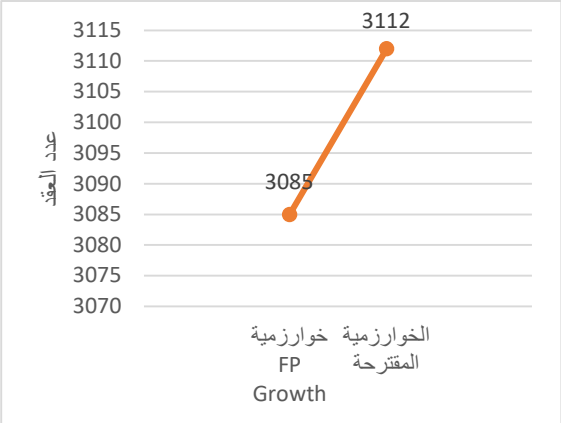
تم تطبيق خوارزمية FP Growth والخوارزمية المقترحة على الجدول Verb_Noun لتتالي فعل واسم والمؤلف من 3000 سطر، علماً انه تم تنفيذ الخوارزميات بلغة الجافا، على حاسب بالمواصفات التالية: المعالج intel core i7-2.20GHz والذاكرة 8GB ونظام Windows 7، فكان زمن تنفيذ الخوارزمية وعدد العقد في شجرة FP tree كتالي:

الجدول 9: نتائج تنفيذ خوارزمية FP Growth والخوارزمية المقترحة

الخوارزمية	زمن التنفيذ	عدد العقد
خوارزمية FP Growth	232 ms	3085 عقدة
الخوارزمية المقترحة	144 ms	3112 عقدة



الشكل6: مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة من حيث زمن التنفيذ



الشكل7: مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة من حيث عدد العقد في الشجرة

بالمقارنة فإن الخوارزمية المقترحة أسرع من خوارزمية FP Growth، لأنها تمر مرة واحدة على قاعدة البيانات وبذلك قد تم اختصار الزمن اللازم لتنفيذ الخوارزمية، لكن نلاحظ أن عدد العقد في الشجرة التي تم الحصول عليها من الخوارزمية المقترحة أكبر من عدد العقد في الشجرة الناتجة عن خوارزمية FP Growth ، وذلك لأن ترتيب الكلمات المتكررة ترتيباً تنازلياً بحسب قيمة تكرارها في خوارزمية FP Growth يؤدي الى الحصول على شجرة افضل (شجرة بأقل عقد)، بينما تغيير هذا الشرط في الخوارزمية المقترحة لا يعطي دائماً أفضل شجرة ولا يعطي العدد الأقل من العقد، بالتالي التحسين الذي تم القيام به للحصول على قواعد ترابط تراعي تسلسل ورود الكلمات، أدى الى عدم الحصول على الشجرة الافضل، فكان لابد من القيام بتعديل ثاني على آلية بناء الشجرة من أجل تقليل عدد العقد في الشجرة بالتالي تقليل المساحة التخزينية اللازمة لتخزين الشجرة.

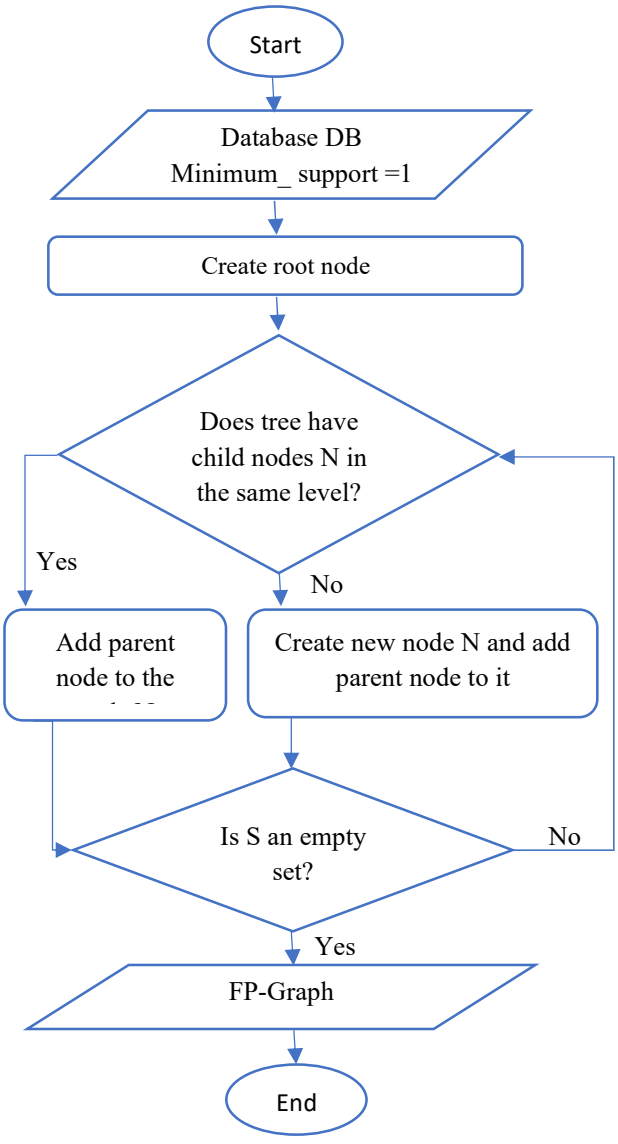
4.6 الخوارزمية المقترحة بالاعتماد على المخطط graph:

ان خوارزمية FP Growth تبني الشجرة بطريقة تجعل عقد المستوى الأول في شجرة FP tree لا تتكرر، بينما يوجد احتمال لتكرار العقد في المستوى الثاني وما بعده، لذلك تم اقتراح تحسينات على طريقة بناء الشجرة لتقليل عدد العقد، بحيث تبقى خطوات بناء الشجرة كما هي في الخوارزمية المقترحة مع إضافة الشرط التالي عند انشاء فرع في الشجرة لكل سطر من أسطر قاعدة البيانات:

إذا كانت الشجرة تتضمن عقدة في المستوى نفسه لها الاسم نفسه للعنصر الحالي في قاعدة البيانات لا يتم انشاء عقدة جديدة بل يتم ربطها مع العقدة الاب والا يتم انشاء عقدة جديدة وربطها مع العقدة الاب لها.

نلاحظ أن التعديل الذي طرأ على بنية الشجرة هو أن العقدة يمكن أن يكون لها أكثر من اب واحد، بالتالي البنية لم تعد شجرة لان الشجرة تعرّف على انها هي بنية بيانات هرمية غير خطية، ولكل عقدة اب واحد فقط و لا يجوز أن يكون للعقدة أكثر من أب واحد، ولا يمكن تشكيل حلقات في الشجرة، [16] لذلك أصبحت البنية عبارة عن مخطط graph وهو بنية بيانات غير خطية تتكون من مجموعة من العقد المتصلة بواسطة روابط، يرمز له بالشكل G(V,E) حيث Vertices: V هي مجموعة العقد ، و Edges:E هي مجموعة الروابط (الحواف) التي تربط بين عقدتين، وهناك أنواع مختلفة للمخططات، لكن النوع الذي تم الحصول عليه من الخوارزمية السابقة هو DAG (directed acyclic graph) مخطط موجه بدون حلقات.

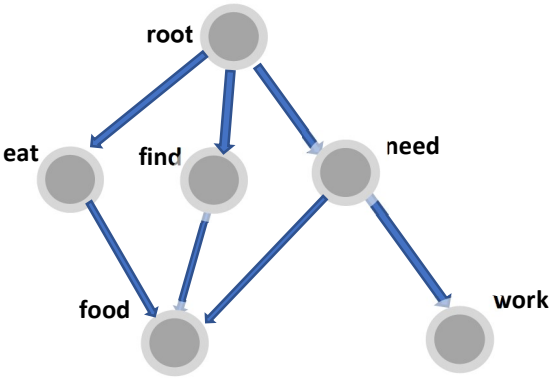
الشكل التالي يمثل المخطط التدفقي لانشاء المخطط graph:



الشكل8: المخطط التدفقي لانشاء المخطط graph

4.6.1 مثال عملي على الخوارزمية المقترحة بالاعتماد على المخطط graph:

سيتم تطبيق الخوارزمية المقترحة بالاعتماد على المخطط من أجل المثال السابق نفسه في الجدول (6)، بالتالي سيتم المرور على الجدول مباشرة لبناء المخطط الموضحة بالشكل(5) كما يلي:



الشكل9 : المخطط graph بعد تطبيق الخوارزمية المقترحة بالاعتماد على المخطط

من الملاحظ في الشكل (5) عدم ورود تكرار لأي كلمة في عقد المستوى الثاني، ليصبح عدد العقد في المخطط ست عقد بدلاً من ثمان عقد كما في الخوارزمية المقترحة وبذلك قد تم تقليل عدد العقد.

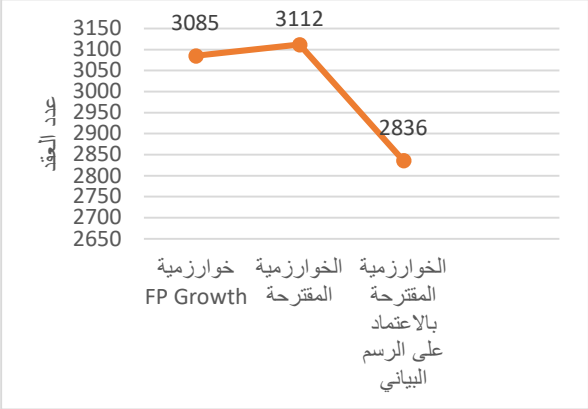
4.6.2 مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة والخوارزمية المعتمدة على المخطط:

تم تطبيق خوارزمية FP Growth والخوارزمية المقترحة والخوارزمية المعتمدة على المخطط على الجدول Verb_Noun لتتالي فعل واسم مؤلف من 3000 سطر، فكان زمن تنفيذ الخوارزمية وعدد العقد وعدد الروابط بين العقد كتالي:

الجدول 10: نتائج تنفيذ خوارزمية FP Growth والخوارزمية المقترحة والخوارزمية المعتمدة على المخطط

الخوارزمية	زمن التنفيذ	عدد العقد	عدد الروابط بين العقد
خوارزمية FP Growth	232 ms	3085 عقدة	3084 رابط
الخوارزمية المقترحة	144 ms	3112 عقدة	3111 رابط
الخوارزمية المقترحة بالاعتماد على المخطط	198 ms	2836 عقدة	3111 رابط

الشكل10: مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة والخوارزمية المعتمدة على المخطط من حيث زمن التنفيذ



الشكل 11: مقارنة بين خوارزمية FP Growth والخوارزمية المقترحة والخوارزمية المعتمدة على المخطط من حيث عدد العقد في الشجرة

بالمقارنة نلاحظ أن عدد العقد في الخوارزمية المقترحة بالاعتماد على المخطط هو الأقل لأنه تم الغاء التكرار في العقد لكل مستوى من مستويات الشجرة، بالتالي الذاكرة اللازمة للتخزين اقل، كما نلاحظ أن زمن تنفيذ الخوارزمية المقترحة بالاعتماد على المخطط أكثر من زمن تنفيذ الخوارزمية المقترحة، لان الآلية الجديدة لبناء المخطط تستغرق وقتاً أكثر لكنه يبقى اقل من زمن تنفذ خوارزمية FP Growth.

في الخوارزمية المقترحة نتجت شجرة تحتوي 3112 عقدة و3111 رابط بين العقد وفي الخوارزمية المعتمدة على المخطط نتج مخطط يحتوي على 2826 عقدة و3111 رابط أي نفس عدد الروابط في الشجرة والمخطط، لذا ستتم المقارنة بين سرعة البحث في الشجرة وفي المخطط باستخدام خوارزميتي بحث هما Breadth First Search (BFS) و Depth First Search (DFS) حيث:

DFS: في هذه الخوارزمية يتم اختيار عقدة مصدر (عقدة الجذر) ليبدأ البحث منها ثم يتم اكتشاف العقد المجاورة على طول كل مسار، أي يتم الانتقال عمودياً باتجاه العمق حتى يتم إيجاد العقدة الهدف او يتم الوصول الى عقدة ليس لها أبناء عندها يتم التراجع backtracking نحو العقدة الاحدث التي لم يتم اكتشافها بعد وهكذا تتكرر العملية حتى الوصول الى العقدة الهدف، علماً ان بنية البيانات التي يتم استخدامها في هذه الخوارزمية هي المكدس Stack [16]

BFS: في هذه الخوارزمية يتم اختيار عقدة مصدر (عقدة الجذر) ليبدأ البحث منها ثم يتم اكتشاف جميع العقد المجاورة (+العقد المتصلة مباشرة بالعقدة المصدر)، أي يتم الانتقال افقياً لزيارة جميع عقد المستوى الأول ثم الانتقال الى المستوى التالي وهكذا تتكرر العملية حتى الوصول الى العقدة الهدف، علماً ان بنية البيانات التي يتم استخدامها في هذه الخوارزمية هي الرتل Queue [16]

فكان زمن البحث في كامل الشجرة والمخطط بالنسبة للخوارزميتين كتالي:

الجدول 11 : نتائج تنفيذ خوارزمية DFS وخوارزمية BFS على المخطط والشجرة

زمن البحث	خوارزمية BFS	خوارزمية DFS
زمن البحث في الشجرة	8ms	12ms
زمن البحث في المخطط	6ms	9ms

نلاحظ ان خوارزمية BFS أسرع من DFS لان الأخيرة تعاني من مشكلة الرجوع الى الوراء backtracking المتكرر في حال الفشل والتتبع في الشجرة او المخطط بالتالي زمن تنفيذ أكبر، كما نلاحظ ان زمن البحث في المخطط اقل من زمن البحث في الشجرة، لذلك سيتم الاعتماد على الخوارزمية المقترحة بالاعتماد على المخطط وسيتم تنفيذ هذه الخوارزمية على الجداول الخمسة لقواعد المعنى التي تم ذكرها سابقاً في الفقرة (4-1)، للحصول على قواعد المعنى المطلوبة لاستخدامها في التحقق من صحة معاني جمل اللغة الإنكليزية.

وفي مايلي جدول مقارنة للخوارزمية المقترحة بالاعتماد على المخطط مع عدة خوارزميات تنقيب أخرى، عند تنفيذها على مثالنا في الجدول (6):

الجدول 12 : مقارنة بين الخوارزمية المقترحة بالاعتماد على المخطط وخوارزميات تنقيب اخرى

الخوارزمية / الصفات	Apriori	Vertical data forma	FP Growth	الخوارزمية المقترحة بالاعتماد على المخطط
مسح قاعدة البيانات	عدة مرات	عدت مرات	مرتين فقط	مرة واحدة فقط
التقنية	Breadth first search	Depth first Search	فرق تسد	فرق تسد
بنية التخزين	Array	Array	Tree	Graph
الزمن	12 ms وقت تنفيذ كبير	9 ms وقت تنفيذ اقل من خوارزمية Apriori	7 ms وقت تنفيذ اقل من خوارزمية Apriori	5 ms وقت تنفيذ اقل من خوارزمية FP Growth
العيوب	مجموعة العناصر المرشحة كثيرة جداً والمرور المتكرر على قاعدة البيانات يتطلب مساحة ذاكرة كبيرة	تتطلب ذاكرة إضافية لتخزين القوائم الوسيطية التي تضم ارقام أسطر العناصر	شجرة FP tree مكلفة في البناء وتحتاج الى مذيد من ذاكرة	لاتعمل الا مع قيمة دعم اصغري تساوي الواحد

5. الاستنتاجات والتوصيات

5.1 الاستنتاجات:

تم في هذا البحث تنفيذ خوارزمية FP Growth بلغة الجافا لإيجاد قواعد المعنى، ونظراً لأن هذه الخوارزمية تعطي قواعد ترابط تتجاهل تسلسل ورود الكلمات ضمنها، تم انجاز تعديلات على الخوارزمية فتم الحصول على الخوارزمية المقترحة بالاعتماد على المخطط التي تعتبر خوارزمية فعالة في مجال تحليل المعنى، لأنها تعطي قواعد ترابط تراعي تسلسل ورود الكلمات

ضمنها والذي يعتبر امراً مهماً في تحليل المعاني، كما انها أسرع من خوارزمية FP Growth في إيجاد قواعد المعنى، وتحتاج الى ذاكرة اقل للتخزين.

وتم استخدام قواعد المعنى الناتجة عن الخوارزمية في التحقق من صحة معاني جمل اللغة الإنكليزية، بالتالي تم في هذا البحث تقديم نموذج لتوليد قواعد المعنى، يعتبر نظاماً قابلاً للتعلم لأنه من الممكن تحديث وإعادة بناء هذا النموذج عند توفر بيانات جديدة مضافة إلى قاعدة البيانات، وهذا النموذج يجعل من الحاسب أداة تحاكي الانسان الخبير في مجال اللغة الإنكليزية ويفتح آفاق مستقبلية لاستثماره في مجال اللغة العربية.

5.2 *التوصيات:*

- احداث مخابر لغوية تتضمن لغويين ومختصين في علم الحاسب.
- استخدام خوارزميات التنقيب في المعطيات في مجال معالجة اللغات الطبيعية.

المصادر

[1] Bender, E. M. (2013). Linguistic Fundamentals for Natural Language Processing: 100 Essentials from Morphology and Syntax. Synthesis Lectures on Human Language Technologies, 6(3), 184 pages.

[2] Bhujade, V., & Jonwe, N.J. (2011, October) . Knowledge Discovery in Text Mining Technique Using Association Rules Extraction. in 2011 International Conference on Computational Intelligence and Communication Networks (pp. 498-502). IEEE.

[3] Daves, M. (2011). N-grams data from the Corpus of Contemporary American English (COCA), Downloaded from http://www.ngrams.info on May 04, 2020.

[4] Guo, Y., & Wang, Z. (2010, March). A vertical format algorithm for mining frequent item sets. 2010 2nd International Conference on Advanced Computer Control (pp.11-13). IEEE.

[5] Han, J., & Kamber, M . (2006) .Data Mining: Concepts and Techniques. 2nd ed, Elsevier Inc, United States of America,772 pages.

[6] Han, J., & Pei, J. (2000). Mining Frequent Patters by Pattern Growth: Methodology and Implications, ACM SIGKDD Explorations Newsletter, 2(2), 30-36.

[7] Luo, X., & Chen, Z .(2020). English text quality analysis based on recurrent neural network and semantic segmentation. Future Generation Computer Systems, vol.112 , 507–511.

[8] Medjahed, B., & Bouguettaya, A. (2005) . A Multilevel Composability Model for Semantic Web Services. IEEE Transactions on Knowledge and Data Engineering, 17(7) ,954 – 968.

[9] Moraes, S., Godbole, A., & Gharpure, P. (2017, September). Affinity analysis for context-free grammars. in 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI).

[10] Motoda, H., & Ohara, K. (2009). The Top Ten Algorithms in Data Mining, Taylor and Francis Group, LLC ,United States of America, 214 pages.

[11] Omar, N., Hanna, P., & Mc Kevitt, P. (2006, June). Semantic Analysis in the Automation of ER Modelling through Natural Language Processing. In 2006 International Conference on Computing & Informatics . IEEE

[12] Osipov, G., Smirnov, I., Tikhomirov, I ., & Vybornova, O . (2012, September). Technologies for Semantic Analysis of Scientific Publications. In 2012 6th IEEE International Conference Intelligent Systems. IEEE.

[13] Singh, A., Agarwal, J., & Rana, A. (2013).Performance Measure of Similis and FP-Growth Algorithm,International Journal of Computer Applications,Volume 62– No.6.

[14] Singh, A. K., Kumar, A., & Maurya, A. K. (2014, May). An empirical analysis and comparison of apriori and FP- growth algorithm for frequent pattern mining. 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp.1599-1602). IEEE.

[18] Yamuna Devi, N., & Devi Shree,J. (2013). A novel approach and comparative study of association rule algorithms in validation of semantics of sentences, International Journal of Computer Applications ,France, Vol 62 , No 3 , 22-26.

[15] Vij, S., Jain, A., Tayal, D., & Castillo, O. (2017). Fuzzy Logic for Inculcating Significance of Semantic Relations in Word Sense Disambiguation Using a WordNet Graph. International Journal of Fuzzy Systems , 20(2) , 444 – 459.

[16] Welborn, C., & Rudolph, G. (2020, October). Formal Definitions for Common Data Structures and Algorithms. in 2020 Intermountain Engineering, Technology and Computing (IETC).

[17] Witten, L., Frank, E., & Hall, M. (2011). Data Mining Practical Machine Learning Tools and Techniques, 3th ed, Elsevier Inc, United States of America,665 pages.

The Semantic Analysis of English Language Sentences by Using the Improved Data Mining Algorithms

***Miriam Obied*[1,*], *Yaroub Dayoub*[2]**

[1]*Department of Information Technology, Faculty of Information and Communication Technology Engineering, Tartous University, Syria.*

[2] *Department of Information Technology, Faculty of Information and Communication Technology Engineering, Tartous University, Syria.*

, yaaroub-dayoub@hpu.edu.sy

**Corresponding author:Miriam Obied, miriamobied@hotmail.com*

Published online: 30 September 2022

Abstract— Proofreading the meanings of the natural languages is considered one of the main goals of linguists and those who are interested in computational linguistics. That is because it has become essential to check written texts on a computer in different areas..This paper presents a model for validate the meaning of English sentences through generating semantic rules from a database which includes the most recurrent words in the English language, based on one of the data mining algorithms, which is FP Growth algorithm.This algorithm generates association rules between words, but those rules which result from the FP Growth algorithm ignore the sequence of words which is considered an important issue in semantic analisis. That is why the algorithm has been modified in order to get association rules which give importance to the sequence of words, taking into consideration the time needed to get those rules and the memory needed to store them.

Keywords— Semantic Analisis, Data Mining, Semantic Rules, FP Growth algorithm, Association Rules.